Train, store, analyse for more adaptive teaching

Agathe Merceron, Kalina Yacef

Train, store, analyse for more adaptive teaching

Agathe MERCERON and Kalina YACEF

ESILV - Pôle Universitaire Léonard de Vinci, France
Agathe.Merceron@devinci.fr
School of Information Technologies - University of Sydney, Australia, kalina@it.usyd.edu.au

Résumé

Il existe maintenant un nombre important d'outils tuteurs qui permettent aux apprenants de pratiquer leurs compétences dans un domaine donné. Un système idéal comprendrait une plate-forme d'apprentissage pour les étudiants, mémoriserait leurs réponses et les diagnostics de leurs erreurs, et aiderait l'enseignant à analyser ces données. Cette communication présente notre vue d'un tel système qui permet aux apprenants de s'entraîner, qui mémorise des informations riches recueillies auprès des apprenants et qui a des facilités pour analyser et extraire des informations pertinentes d'un point de vue pédagogique de ces données de manière à permettre aux enseignants d'adapter leur enseignement. Nous montrons comment un tel système est en partie déjà réalisé au travers de deux outils, le Logic-ITA (entraînement, mémorisation) et TADA-Ed (analyse) et comment les enseignants peuvent les utiliser pour adapter leur enseignement et ouvrir des perspectives pour améliorer le système ?

Mots-clés : Training, aides intelligentes.

Abstract

There are now a large number of web-based tutoring tools which allow students to practice their skills and knowledge. An ideal system would provide a training platform for students, store their answers and diagnosis of their mistakes, and help teachers analyse the data. This paper presents our view of such a system that provides training to learners, stores rich data from learners, has facilities to analyse and extract pedagogically relevant information from these data so that teachers can adapt their teaching accordingly. We show how such a system is already partially working with two tools, the Logic-ITA (train, store) and TADA-Ed (analyse), how teachers can use this system to adapt their teaching and give perspectives to improve the system further.

Keywords: Tutoring, intelligent support.

Introduction

Web-based educational technologies are becoming more and more important. In particular, there are now a large number of web-based tutoring tools which allow students to practice their skills and knowledge. Whilst they are very attractive to students because of the time, geographical and content flexibility they offer them, they also provide teachers and researchers with the opportunity to study learning patterns in a grand scale and have a greater insight into students' learning difficulties. This is also particularly useful for cases where face-to-face interactions between teachers and students are scarce (i.e. large classes or online education). Being tutoring tools, students typically execute exercises, solve problems or manipulate a simulated environment. They receive feedback, explanations, help and hints whilst they practice. These benefits to the learners are extremely valuable. In parallel, these systems collect, potentially at least, very large quantities of data about each user such as all the steps entered by the student including the mistakes made while solving exercises. Having electronic access to rich student answers (not only yes/no answers) makes it possible to analyse them, extract pedagogically relevant information and provide feedback to the students and to the teacher. It becomes even possible to provide some automatic monitoring on the progress made by students to teachers.

Analysis presents two difficulties. First, the access to this huge electronic base of information is not a solution in itself. Well-defined ways to exploit the data from a pedagogical perspective need to be found. The field of data mining is precisely concerned with finding patterns in vast quantities of data, and we see a recent emerging of its use in a pedagogical context. Some researchers use classification techniques to predict student performance fairly accurately [1]. Other systems analyse web logs of student use to infer behavioral information for teachers [2 ; 3]. In the case of web-based ITS, the data is semantically richer and can lead to more diagnostic analysis, such as finding mistakes often associated together [4]. Since the process is relatively new, there is not yet a great deal of studies made by education researchers trying to extract information from such data. The mining that can be performed is endless. The usefulness of mining such data is promising but still needs to be proven and stereotypical analysis to be streamlined. Secondly, teachers (unless it is their field of expertise) are not all computer experts, let alone data mining experts. Tools to extract data need to have an intuitive interface to be easy to use by teachers and good visualization facilities to make their results meaningful to teachers.

Though difficulties are not all solved, this paper advocates for such learning systems that are enhanced with analysis tools so that the resulting analysis has the potential to help teaching and learning. In section 2, the overview of such a system and its benefit for education is sketched. Section 3 gives the roots for our vision, a web-based tutoring tool called the Logic-ITA and a Data Mining tool called TADA-Ed. The Logic-ITA constitutes the train and store components while TADA-Ed constitutes the analysis component. In particular, some analysis of students' answers and the use of its result to improve teaching are presented. Based on these experiments, section 4 presents some perspectives to improve analysis.

Train, Store, Analyse, Adapt in the educational loop

We advocate the usefulness of a process illustrated in Figure 1. Three technological components, namely train, store and analyse, have a role and there are two types of users: the teachers and the students. The web-based tutoring tool is the TRAIN component that presents course material to students and allow them to practice exercises. The STORE component collates every detail of students' works in a database. Students' works may include the type of exercises they have made and their underlying concepts, the students' answers, their mistakes, the diagnosis of the mistakes, the time and date. As said above, keeping track of whether answers are wrong or right only is not enough to provide teachers with information that is relevant enough for teaching purposes. The ANALYSE component allows to extract pedagogically relevant information for both learners and teachers which then enables users (teachers or learners) to ADAPT in a suitable way their learning or teaching. In this paper we focus on the information that can be extracted to help teachers improve teaching.

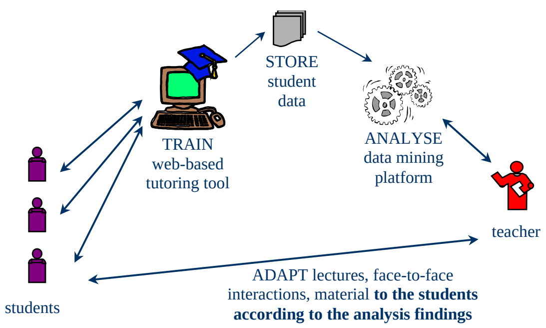

Figure 1- Educational loop: using technology to enhance teaching

The Analyse component may provide the following feedback:.

- Find the most frequently made mistakes.
- Find the exercises that everybody got right (or wrong).
- Find all students who have not been able to complete successfully any of the exercises that they have attempted.
- Cluster students by abilities, for example cluster students in 3 groups: those with problems with concepts X and Y, those who are on top of things, and those who hardly did anything on those concepts.
- When enough data have been collected, then it is possible to build a decision tree classifying students who pass and fail according on how they did in their training. For the teacher, it is possible to get information like "student X is likely to fail the course if does not pick up".
- Find mistakes often associated together like "if students make mistakes A, B then they make also mistake C ".

It is clear that this information related to the current class is useful to teach better. Being more aware of how a class is going and responding to the topics mean that teachers can be more adaptive. For example:

- Teachers can be pro-active with students with difficulties, for instance those who have not been able to complete successfully any of the exercises that they have attempted.
- Being aware of mistakes associated together can lead them to review the way they present parts of the material to students.
- Finding out a set of clusters to partition the class by ability/difficulty can be used to provide different material and level of help to students according to their cluster.

Such information can also be used to support teachers in their teaching and follow-up of students. For example, it can be the case that making mistakes A, B and C is part of the normal learning process. As long as only these associations are detected, nothing needs to be done. However, if something unexpected comes up, teachers should be notified. The analysis part of the system could have an 'alert' component that makes teachers aware of potential problems.

Progress so far

We are working towards building a generic platform that teachers can use to analyse any kind of student data captured from a computerised learning tool. This platform would have the autonomy to perform routine analysis and generate a report for the teacher as well as allow the teacher to perform more specific analysis. It would highlight unexpected or unusual patterns so that the teacher can perform further analysis. It would adapt to the teacher and provide a power of analysis matching the teacher's level of data mining expertise. For this we need to (i) identify the patterns that are likely to be of interest to researchers and teachers, (ii) identify the techniques that are likely to exhibit interesting patterns and how to use them and (iii) create an interface that does not overload the teacher yet that makes powerful use of visualisation to detect patterns.

Our approach to aim for this ideal system is *iterative* and *empiric*. It takes its root in our experience with the Logic-ITA and TADA-Ed. The Logic-ITA is a web-based intelligent teaching assistant comprising a practice tool for the students and the collation of students' works into a database. We have designed a platform Tool for Advanced Data Analysis for Education, TADA-Ed, which contains some data mining algorithms as well as visualisation graphs tools. Let us now describe the four stages of the process in this context.

Train

The Logic-ITA helps students practice formal proofs whilst keeping the lecturer informed of how students progress. The Logic Tutor is the training component of the Logic-ITA and has been in use since 2001 in our school. As part of an undergraduate course, students need to learn and understand the concept behind formal proofs of logic. But they do not have enough opportunities to practice with feedback. The Logic Tutor helps them to do this. When they use it, students can choose to create new exercises, select exercises in the exercise database, or ask the system for one adapted to their needs. The system stores, for each student, every step entered by the student, along with any mistake they may have made. We will not describe in great details the whole system (the reader can refer to [5] but we need to explain how some of this data is generated to make the following sections clearer.

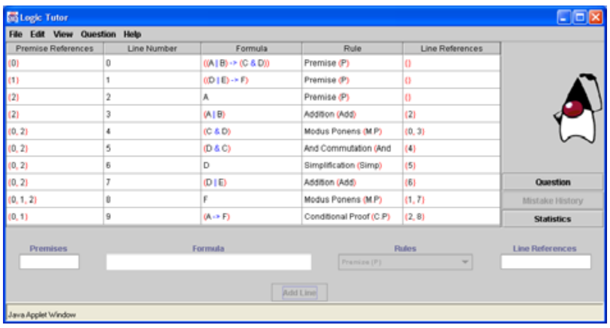

Figure 2 . Screen shot of a completed exercise

Exercises start with a given set of premises, i.e. a set of well-formed formulae (wff) of propositional logic, and exactly one wff, the conclusion. The task then consists of deriving the conclusion from the premises, step-by-step, using laws of equivalence and rules of inference (we will refer to both of these as rules for the rest of this paper).

Figure 2 shows an exercise. Here the student is given the first two lines (lines 1 and 2) and the conclusion to derive, (A -> F). For each step, the student must fill out a new line. The student needs to enter the following information in the bottom pat of the screen:

- enter a formula in the *Formula* section,
- choose, from a pop-up menu, the rule used to derive this formula from one or more previous line(s) (*Rules*),
- the references of those previous lines (*Line References*) and
- the premises the formula relies on (*Premises).*

For example in Figure 1, the last line is currently deriving the formula (A -> F), using the rule Conditional Proof and the formulae of lines 3 and 9. Because lines 3 and 9 rely respectively on premises {3} and {1,2,3} (as can be seen in the first column) and *Conditional proof* removes premise 3 (the extra premise that has been added), the line entered therefore relies on premises {1,2}. It is actually the last step of this exercise, deriving the conclusion.

There are often many ways to prove an argument valid. The actual path followed is not important, as long as each step is valid. Students here have total freedom in the reasoning they choose to follow.

At each step, the system checks the validity of the data entered by the student. There are different types of mistakes, and, each of them is labelled with a meaningful title for the teacher. Some are generic: *Wrong reference lines*, and others are more specific *Simplification before Commutation*. In addition, the rule specified at the time of the mistake is also recorded and linked to the mistake.

Store

Each student data is stored in an individual file and is retrieved whenever the student logs in again. That way, the system knows about the student's history, mistakes, and is able to choose exercises tailored to the students' needs. At any time the teacher requires it, the Logic-ITA collates all the information stored in the student files by the Logic Tutor into a database. The information that is stored is not merely whether a student has solved successfully an exercise or not. For each student, it is possible to retrieve all the exercises attempted and their level, the date they have been done, along with all the mistakes made and the logical rules correctly used for each exercise attempted. For each exercise attempted, a performance is calculated according to whether the exercise has been solved, whether a shortest path to solution has been found, and to whether mistakes were made or not. Using performance, a level is calculated for each student from 1 (lowest) to 5 (highest).

Analyse

Currently, analysis is done by two means. First, graphics and simple queries on the database help the teacher follow what happens. For example, the database is queried to retrieve simple information such as the logical rules causing the most mistakes. This is integrated in the Logic-ITA and already provides useful feedback to the teacher. Second, data mining techniques allow to look for more hidden patterns and visualize them in an intuitive way. We built a tool, called TADA-Ed, which we present briefly here (more details can be found in [6]).

Figure 3 shows a screenshot of the main interface of TADA-Ed. The two top windows are for visualising the data or the results of some data mining algorithm in two dimensions, according to any given criteria. The two bottom windows are dedicated to clustering (presently k-means and hierarchical clustering), classification (decision tree C4.5) and association rules (a priori algorithm). The left windowpane shows the detail of any data selected by the user. All windows are linked together: for example the results of k-means is displayed in the points view using a colour for each cluster, the selection of a subset of elements in the chart (left top) is displayed in red in the right color and its details are displayed in the message area.

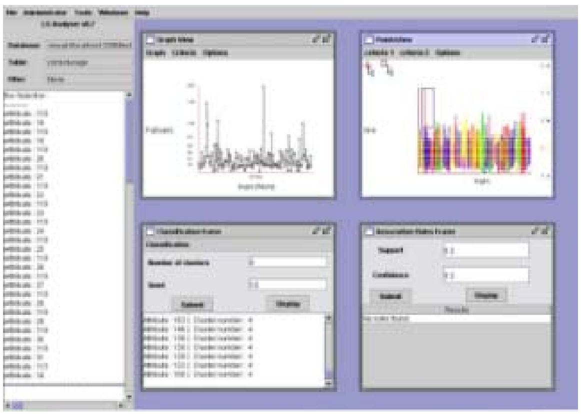

Figure 3. Screen shot in Tada-Ed

Mining association rules

While mining association rules, the pattern that one is looking for is items often occurring together. In our context, these items are mistakes. The purpose of looking for mistakes often occurring together is for the teacher to emphasise subtleties while explaining formal proofs to students.

Association rules have the form: A, B -> C, *support* 65%, *confidence* 86%.
This reads as follows. If students make mistakes A and B while doing an exercise, then they make also mistake C with a support of 65% and a confidence of 86%. This means that mistakes A, B and C occur together in 65% of all exercises attempted (according to the support). Confidence measures how much C is really implied by A, B. The higher the value, the greater the dependency is. When mining the database of the Logic-ITA, the four mistakes often found associated together are:

Rule can be applied, but deduction incorrect,
Premise set incorrect,
Wrong number of line references given and
Incorrect line numbers.

Before we give an overview of the associations found, we illustrate what they mean, referring at the example shown in Figure 2. Consider line 3. If the student entered the formula (B | A) instead of (A |B), the mistake *Rule can be applied, but deduction incorrect* would be made. Indeed, *Addition* can be applied, but the added proposition comes last, as shown in Figure 2, not first as written here. Consider now line 4. Suppose the student gives only 2 in the *Prem.* field. Then the mistake *Premise set incorrect* would be made. Suppose now the student gives only 0 in the *References* field. Then a *Wrong number of line references given* mistake would be made, because 2 lines of reference are needed. Finally, suppose the student gives the line numbers 0 and 1, then a *Incorrect line numbers* mistake is made, because the second line entered is wrong.

Rule can be applied, but deduction incorrect	Wrong number of line references given	Premise set incorrect		Rule can be applied, but deduction incorrect	Wrong number of line references given	Premise set incorrect	**Support**	**Confidence**
X			→		X		70%	90%
	X		→	X			70%	80%
X			→			X	65%	84%
		X	→	X			65%	82%
	X		→			X	70%	83%
		X	→		X		70%	88%

Table 1Association rules for Year 2001 with a support higher than 65%.

Although we cannot include here all the rules obtained for each year, we can say that the results show a great regularity between the years 2001 -2003. Always the same mistakes are made together, except for *Incorrect line numbers* that was present with a lower support (around 40%) in 2001 and 2002. Also supports and confidence are quite comparable, though supports are a little less for the year 2002. These associations show relationships between mistakes involving line numbers in the *Premises (Premise set incorrect*), line numbers in the reference lines a logic rule applies to (*Wrong number of line references given, Incorrect line numbers*) and incorrect formulas (*Rule can be applied, but deduction incorrect*), which all constitute the core of logical proofs. This confirms what human tutors had sensed. Students often have difficulties at grasping the formal side of proofs and include exactly all the details required in a proof: one has to provide not only a correct formula and a logic rule, but also the lines it applies to, and these are different from the premises involved. The tool forces students to pay attention to all its details.

Clustering

When students make mistakes but finish successfully exercises, teachers do not need to worry. However, teachers need to be aware of students not completing successfully exercises since they may have difficulties. In order to characterize these students, a clustering method called k-means clustering has been applied, taking into account the recorded mistakes and the exercises attempted. The clustering for the year 2002 yielded three classes. Class 1 is composed of students making few mistakes, class 2 of students making an intermediate number of mistakes and class 3 students making many mistakes, see [6] for more details. Then we have visualised the results in various graphs. When visualising the various exercises attempted by each student, we found that the trend was that students of class 1 attempt more exercises than students from class 2 or 3. When displaying *student logins* against *mistakes*, we found that the trend was that students from class 2 or 3 make more different kinds of mistakes than students from class 1. Students from class 1 make the mistakes that are most usually made by everybody using the Logic-ITA. Plotting *student logins* against the *logic rules* used in the non-completed exercises gave the graph given in Figure 3. This graph shows vertical lines for several students from class 2 or 3 only, not from class 1. Students from class 1 constitute the narrow green strip in the middle of the graph. A vertical line means that all rules have been tried while doing the exercises. They suggest to us that these students have just tried one rule after the other from the pop-up menu, apparently adopting a behaviour of "guess and test" strategy. However, this interpretation needs to be checked.

Adapt

The positive correlation between the use of the Logic Tutor and the exam results shows that 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 [4]. Nonetheless, we have used the information found from analysing the student data to provide a little more adaptive teaching.

Re-focus the contents of revision lectures

The results of the LT-Analyser queries was primarily used to re-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.

Improve the course

After mining the mistakes often associated together for the years 2001 and 2002 [4], modifications have been provided to the course.

a) The concept itself of formal proofs, which seems to be often misunderstood, was introduced earlier in 2003. Also, the fundamental difference between laws of equivalence and rules of inference (a possible cause for the mistake formulas *Rule can be applied, but deduction incorrect)* was 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 [7]. 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.

These changes have led to better-structured lectures. The exam results on the question on logic formal proofs using the natural deduction system are steadily improving throughout the year (average of 4.3 (sd=2.7), 4.7 (sd=2.8), 4.9 (sd=2.8) for the years 2001, 2002 and 2003 respectively compared with 3.3 (sd=1.6) in 2000 where students did not use the tool at all). Curiously however, looking at the results obtained for 2003, they do not seem to have any consequence in the mistakes often made together. There is still the same level of association between the mistakes mentioned above. A reason could be that students use the tool to experiment, to learn and train themselves, not to do as little mistakes as possible. Therefore, we take as a matter of fact that learning goes together with making mistakes and correcting them and we recognize these 4 types of mistakes as part of the learning phase. A way to adapt better could be to implement some automatic monitoring during the course that would alert the teacher only for mistakes that are made often and that are different from the ones already mentioned.

A similar clustering has been made with the data of the year 2003 and gave a similar result: students making many mistakes use much more different rules. During the year 2004, we plan to check whether our interpretation of “guess and test” strategy is relevant. In case it is, we plan to have some automatic monitoring providing two kinds of alert to the teacher, one for students not completing exercises and making a few mistakes, and the other one for students not completing exercises but making many mistakes.

Perspectives for a better analysis

Our experiments so far with the data from the Logic-ITA have yielded important improvements to the way we teach and also highlighted some important needs for working towards an ideal system.

- In an educational context, the sequentiality of data is very important and needs to be taken into account. For instance, a mistake made by a student on a topic early in the training but followed by a correct handling of the topic is a positive evidence of learning. Whereas a mistake followed by more mistakes on the same topic suggests the contrary. Therefore, several queries and data mining algorithms need a refined version. For example, the simple query ‘find concepts causing the most mistakes’ need the following refinement ‘find concepts causing the most mistakes not corrected by students’. Similarly, the classical association rules algorithm does not take the sequentiality of items inside a transaction into account. A refined version should produce rules as follows. If students make mistake A while solving an exercise, then they make also mistake B later in the same exercise, with a support of 56% and a confidence of 78%. Note that this is different from taking into account temporality. Taking temporality into account produces association rules such as ‘If students make mistake A while solving an exercise, then they make also mistake B while solving another exercise done later, with a support of 56% and a confidence of 78%’. We are currently working on implementing these refinements in TADA-Ed.

- The static analysis of, say students clusters, reflects the current grouping of the students and is useful to adapt teaching material for example. But it is not sufficient to group them according to learning abilities: in order to group students according to their rate of learning (some are quick learners, some are not), we also need to consider their individual trends, i.e. the evolution of their learning over time/exercises. It is not obvious so far how classical clustering has to be changed to take the rate of learning into account.

- There is a need to execute several algorithms in a row, the results produced by one algorithm become (part of) the input for the next one. TADA-Ed needs to be enhanced to make several steps in a row easier for the user.

- Different data coming from other systems will be analysed to help grasp better which features are relevant and should be included in TADA-Ed.

- There is an important risk to overload the teacher with too much information. The amount and depth of analysis that can be made is endless. The richer the data is, the more we can mine! Therefore, there is a need for further investigation. What are the right queries, what are the right data mining algorithms and how to use them properly to gain pedagogical relevant information? How often should they be run? Ideally, regular tasks could be performed automatically and the results compared against patterns. Then the attention of the teacher could be drawn to the relevant findings if they are important and unusual. He would then have more time to perform more specialised analysis. This topic relates to automatic monitoring. In the case of the Logic-ITA automatic monitoring could look as follows. Every week queries on the mistakes most often done, on association of mistakes and clustering of students not finishing attempted exercises would be performed. The system would alert the teacher only for mistakes or association of mistakes not belonging to the four most made mistakes and give the students not completing exercises clustered according to the number of mistakes made.

- The interface of such a tool needs to be studied carefully. The system is only useful if teachers can use it easily. We cannot expect all teachers to understand how data mining algorithms work before they can start analysing data. An interface displaying data mining keywords would certainly be useless to many teachers. For teachers without much non-computer science background, the interface should probably display menus such as “group the students of the class by ability” as opposed to “k-means” and use only a subset of the tools that are relevant to them. Whereas someone with a strong data mining knowledge should be given the full range of algorithms with their technical terms and options. This also raises the need for an adaptive interface for the teacher. For the moment, as we are in an experimenting stage, the interface is good but requires some data mining knowledge.

References

[1] Minaei-Bidgoli, B., Kashy D.A., Kortemeyer G. and Punch W.F. 2003. Predicting student performance: an application of data mining methods with the educational web-based system LON-CAPA. In Proceedings of ASEE/IEEE Frontiers in Education Conference (eds) Boulder, CO, IEEE.

[2] Smail, C. and Hussmann S. 2003. The Implementation and Evaluation of an Individualised, Web-based, Formative and Summative Assessment Software Tool for Large classes. In Proceedings of ITHET'03 87-92 Marrakech, Morroco, IEEE.

[3] Zaiane, O. R. 2001. Web Usage Mining for a Better Web-Based Learning Environment. In Proceedings of Conference on Advanced Technology for Education (CATE'01) 60-64 Banff, Alberta

[4] Merceron, A. and Yacef K. 2003. A Web-based Tutoring Tool with Mining Facilities to Improve Learning and Teaching. In Proceedings of 11th International Conference on Artificial Intelligence in Education, 201-208. F. Verdejo and U. Hoppe (eds) Sydney, IOS Press,

[5] Yacef, K. 2004. The Logic-ITA in the classroom: a medium scale experiment. *To appear in the International Journal on Artificial Intelligence in Education*.

[6] Benchaffai, M., Debord G., Merceron A. and Yacef K. 2004. TADA-Ed, a tool to visualize and mine students' online work. In Proceedings of the International Conference on Computers in Education, December 2004 (ICCE04) to appear Melbourne, Australia,

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