International Journal of Engineering Education Vol. 32, No. 5(A), pp. 1888–1902, 2016
Printed in Great Britain
0949-149X/91 $3.00+0.00

Metrics to Facilitate Automated Categorization of Student Learning Patterns while using Educational Engineering Software*

LINDA STERN[1], COLIN BURVILL[2], JOHN WEIR[3] and BRUCE FIELD[4]
[1] Department of Computing and Information Systems, The University of Melbourne, Melbourne, Australia.
E-mail: stern@unimelb.edu.au
[2] Department of Mechanical Engineering, The University of Melbourne, Melbourne, Australia.
[3] Consulting Engineer, Australia.
[4] Consulting Engineer, BW Field Pty Ltd, Australia.

In this paper we describe the use of metrics for analysing student interactions with educational software. We applied this metric-based approach to a class of 200 second-year undergraduate students using an educationally-oriented software simulator to solve specific problems in mechanical engineering design. Our results show that a metric can facilitate categorization of student learning patterns. We suggest how similar metrics could allow automated feedback on learning to both students and educators. Since it is based on numerical analysis and modelling, our approach is particularly well-suited to software used to support teaching and learning in engineering and other mathematically-based disciplines.

Keywords: learning analytics; technology for learning; mechanical engineering education; audit trails

1. Introduction

Online learning and technology-assisted learning have been used increasingly over the last two decades, and have been extended more recently to totally on-line courses. While it is often assumed that student learning is improved using technology, it has been difficult to demonstrate this assumption. Some studies have found that technology can decrease the cost of teaching without decreasing quality [1]. The causal role of technology in any improvements observed in learning has been questioned from the earliest days of multimedia educational technology [2]. A large study of learning in middle school classes found no effect of the technology on learning, and at the same time highlighted the multiplicity of factors that can influence studies that attempt to demonstrate a causal relationship between technology and learning [3].

There remains a need for the educator to obtain feedback about *how* students are using technology in their learning. In some cases, students would also benefit from immediate feedback from the software about their learning. Real-time feedback would be particularly useful in totally on-line courses.

Given that students engaged with educational technology are already using a computer, relevant data can be extracted, recorded electronically, and used for feedback and information. While it can be difficult to demonstrate learning *outcomes*, tracking the *process* of student learning using educational software is certainly possible. A particularly convenient method is to construct log files, which can be analysed to yield data that are useful from an educational point of view [4–6]. While "thinking aloud" protocols have also been used for these purposes, log files have the distinct advantage that they can be collected without intruding on students' thought processes as they are learning.

By observing the learning process directly, we can discover patterns in student learning processes. This knowledge can lead to worthwhile adjustments in pedagogical practice. Real-time analysis of patterns can even help identify appropriate points for active intervention, as for example, in intelligent tutoring systems [7].

Because of the significant amounts of data involved, automation is required to economically discover patterns in student learning. Educational data mining has been used for some time to discover patterns in student activities [8]. Educational Data Mining uses machine learning techniques such as classification, clustering, and text mining, as well as statistical approaches such as regression and correlation. Advantages of Educational Data Mining include the fact that it is objective, automated, and can handle a large amount of data. Additionally, Educational Data Mining allows the discovery of unexpected patterns, leading to new conceptions of how learning is taking place. Disadvantages of Educational Data Mining include the opaqueness of traditional machine learning methods, which are not generally intuitive for the non-computer science educator, and the need for large amounts of input data to achieve any significant degree of accuracy.

In parallel with Educational Data Mining, the use

* Accepted 2 May 2016.

of *analytics* has emerged in higher education practice [9]. Analytics were first developed in the business context, and Academic Analytics were initially used solely in the business end of education. Academic Analytics has been employed to compile data for administrative planning such as for trends in enrolment, and to rationalize the use of facilities and services [10]. Clow [11] has pointed out that it is important that the economic framing characteristic of Academic Analytics be complemented with a concern for learning.

In the new field of *Learning* Analytics, approaches from other areas employing analytics have been used to assess student learning, with the ultimate goal of improving learning outcomes [12, 13]. Whereas Academic Analytics concentrates on administrative data and benefits administrators and funding bodies, Learning Analytics focuses on the learning process and is aimed toward benefitting students and academics [14]. Learning Analytics extends Educational Data Mining techniques, and is therefore congruent with that paradigm, but concentrates more on the sense of the analysis and its usefulness in motivating action [15].

To date, Learning Analytics has primarily been concerned with extracting general features of learning, for example measuring participation in online learning activities, or analysing group performance on online quizzes. Analytics tools often capture data from general platforms such as Blackboard and Moodle and present their output in easily digestible reports that facilitate data-informed decision-making. In contrast to Educational Data Mining, the output from Learning Analytics analysis is intuitive and relatively easy to understand, even for the non-specialist. A disadvantage, however, is that the results are general, and require significant further manual processing to answer specific questions about the student learning process. In contrast to Educational Data Mining, unexpected patterns do not usually emerge from Learning Analytics studies.

For the educator using technology to support teaching and learning, there is a need for an accessible way to analyse the data from student usage in more depth, and preferably in real time. To be generally useful, the analysis needs to be more transparent and intuitive than Educational Data Mining, and more pedagogically focussed than the output from most Learning Analytics. Such pedagogically-focussed analytic data could be used by the educator to better guide students in the use of the software, while real-time feedback could help students use the software to better advantage. This is the need we have addressed in the current study.

We have used a simulation-based piece of educational software for mechanical engineering design problems, which has been built to concentrate on pedagogy, as a representative of the class of simulation-based educational software tools. We have taken a metric-based approach to analysing log files. We have focussed on the process that students use in solving a design problem using this software. We show here how metrics can be developed for the purposes of this kind of analysis. We also show the utility of metrics in flattening complex multi-dimensional numeric data, so that learning patterns can be readily discovered

It has been shown that discovering patterns in student learning processes can lead to worthwhile adjustments in pedagogical practice, while real-time analysis of patterns can help identify appropriate points for active intervention, as for example in intelligent tutoring systems [7]. We have set out to address the question of how tracking student use of the software might be used to discover patterns, and thus to investigate student learning. We hypothesized that it would be possible to identify patterns of student learning behaviour by interrogating log files of software-based learning materials. We further hypothesized that it could be possible to identify these patterns in such a way that they could be detected automatically, and possibly even in real time.

In the course of pursuing this objective, we developed a single metric from a complex, multi-dimensional learning space. We have used this metric to capture the data that are most relevant to learning, and to detect different patterns of student learning. We discuss how the metric can support both manual and automated identification of learning patterns, which can, in turn, supply valuable information to the educator and to the student. While the approach is general, the use of numeric data makes this approach particularly well suited to engineering education.

2. Simulation modules for learning in engineering design (SiMLED)

In this study we have concentrated on simulation-based educational software, using as our exemplar the previously developed software Simulation Modules for Learning in Engineering Design (SiMLED) [16, 17]. SiMLED is a multi-variable modelling tool for several types of mechanical engineering artefacts. The tool accepts a range of input variables and computes appropriate performance (output) variables, based on established mathematical relationships. A student working in SiMLED can change the values of the input design parameters, and observe the effect on the output performance variables.

In the SiMLED *Columns* module, used in solving the task in our study, the six input parameters available to the learner are: cross sectional shape, cross sectional dimensions, length, types of end constraints in two planes, axial load, and material. The student can change the values of these parameters, and can observe the effect of the change on two performance variables, namely Cost, and Factor of Safety (F of S) against buckling in two planes. In the tasks used in SiMLED, as in most tasks in the real world, certain input parameters are fixed, while the student must adjust other parameters to obtain the requisite requirements. A log file records each change in input value that the student makes, along with the time of the change and the new values for the output performance variables. When the student changes the view to look at the values of output variables or to see their effect, this is also recorded in the log file.

SiMLED is described in further detail in the Appendix.

3. Experimental design

Students were given a specific mechanical engineering design problem to solve. The problem was closed, with a single best solution that students were asked to determine.

The problem assigned to participants was specified as follows:

> Using SiMLED to help you, attempt the following design task, aiming for a Factor of Safety of 1.0.
>
> What is the maximum axial force that can be supported by a simply-supported brass column, 2000 mm long, with a solid square cross-section of 50 mm width?

Students were expected to manipulate the input variables within SiMLED so as to find the best solution to this problem. As students manipulate the values of the input variables (Appendix Fig. B), SiMLED shows the values for the output variables in graphical form (Appendix Fig. C), so students can follow their progress towards the best solution.

Prior to commencing the experimental task, students undertook a preliminary task to familiarise themselves with the SiMLED interface. Over the course of one week, 363 log files were generated by 200 students. We pre-filtered the log files to exclude sessions that did not reflect student engagement, such as empty files, where a student had simply logged on and immediately logged off, and used only the remaining log files in the rest of this study. The log files contain values for both the independent design variables and for the dependent performance variables, in addition to other information about the session.

3.1 Development of analysis metrics

In this section we discuss the development of a single overall performance metric that allowed us to follow the students' progress towards a solution and categorize student approaches to the problem.

Because the experimental task in this case has a single known best solution, it was possible to measure the progress of students towards the target solution. In each student session, we recorded the values of input variables, output variables, and which view was on the screen. The output variables are shown in Table 1, along with the SiMLED default values and the correct values when the problem has been solved. The magnitudes of the output variables changed as the students changed the input variables (e.g. material, shape, size and applied load). Values of the students' performance (output) parameters at each time step were recorded for later comparison with the corresponding values from the known single best solution.

At any given point in time, we can calculate an error ratio X/X_0 for each output variable X, which shows how close a student's value for that variable is to the value X_0 in the known best solution. The ratio is 1 when X has approached X_0, giving a logarithm of the ratio equal to 0 at the point of solution. We can track the error ratios for each output variable in the time series separately. The student's progress toward the best solution can be followed by constructing a time series for each dependent variable, where the unit of time is one action that the student takes, *e.g.* one mouse click or one typing step that changes an independent variable or view. On average, each action took students 3 seconds, with a large variance.

An example student session is shown in Fig. 1, tracking the error ratio for each of the six output

Table 1. Output (performance) variables available for scrutiny within SiMLED

Output Variable	Default value	Solution value	Initial Error ratio
Cost	20.69	281.6	0.0735
Factor of Safety	1.57	1.00	1.57
Mass	5.17	42.1	0.123
Slenderness (Transverse)	14.855	138.56	0.107
Slenderness (Frontal)	14.855	138.56	0.107
Applied Force / Cross Section Area	151.58	52.95	2.863

variables. In this particular session, the student completed 223 actions, as shown on the x-axis. The y-axis displays the logarithm of the error ratio for each of the different variables, which all converge to zero at around step 160, where the single best solution for all of the output variables has been achieved. The convergence is not smooth, and it is apparent that in this session some variables converged more quickly than others. After the student reached the correct solution, the dependent variable errors move away from zero. Our observations of students working with SiMLED in the computer laboratory suggest that the most likely explanation for this later divergence away from the best solution is that the student knew they had completed the task and were now interested in exploring the relationship between specific variables (see section 4).

While the tracking of all the independent variables over time does capture the log file information, the multi-dimensional nature of the problem solution means that the data are a bit messy, and leaves us without a unified overall measure of the student's underlying problem-solving strategy or efficacy. Therefore, we were interested in developing a single, combined metric that could be used to follow the student's progress more readily.

We developed an overall measure of a student's progress by adding the absolute values of the logarithm of the six individual error ratios to form an aggregate value which we named E (error). The formula for calculating E is shown in Equation (1):

$$E = \sum_i \left| \log_{10}\left(\frac{X_i}{X_0}\right) \right| \qquad (1)$$

where X_i is the student's value for a particular parameter and X_0 is the value for that parameter in the known solution. E is a general measure and includes all six output variables, even though for some problems, subsets of the output measures might be sufficient to track learner behaviour. At the correct solution, the value for E will be close to 0, *i.e.* less than some small value Δ. For convenience, in the rest of this paper we refer to $E < \Delta$ as $E = 0$.

The use of absolute values of the logarithm of the ratios reflects the notion that a deviation from the target value by a factor of 10 is equally as serious as a deviation by an inverse factor of 0.10. We considered that a student has achieved "the correct solution", when the E value is within a nominated error tolerance of the single best solution. We plotted a second time series, based on the same student session we used to show individual output variable values in Fig. 1, but this time using the value for the combined metric E. This time series is shown in Fig. 2, with the value of the aggregate measure E shown against time, again measured in steps of input variable change. As can be seen in this figure, the use of this combined E value has greatly simplified the visual output, allowing a quick visual inspection. In the case of this particular student, displaying E on the y-axis, instead of all the separate output error values, allows us to quickly see that the student has reached the correct solution, with $E = 0$, around step 160.

We note in Fig. 2 that the value for E goes up again after reaching the correct solution, and becomes greater than 0 starting from step 180. As mentioned earlier for the single outputs (Fig. 1), we hypothesized that this student knew they had solved the problem and was now exploring the effect of changing other input variables. We looked more closely at the log file, in order to see whether our hypothesis was justified. The log file showed, in fact, that after the correct solution had been reached, the student began systematically exploring the effects of changing the values of other variables. Immediately after reaching the correct solution, different *shapes* for the column were explored in nine successive steps, without changing any of the other input variables. Column shapes explored were: T, hollow triangle, solid triangle, hollow rectangle, solid rectangle, solid cylinder, and hollow cylinder. Then, in the next 18 steps, different *materials* were explored, including brass (as specified in the problem) and several types of steel. End constraints (*Supports* controller), which were pre-set for this problem, were also varied, so it was clear that this student was exploring the effect of input parameters on output variables. Comparing Fig. 2 with Fig. 1, we can see how much more easily this sort of exploratory behaviour can be picked up with the aggregate output metric E.

4. Comparing student sessions

Having developed the metric E, and having seen that it could be used to follow a single student's progress in solving a set task (Fig. 2), we next set out to see if we could identify patterns of learning that were more generally common among students. We hypothesized that student approaches to the task might fall into a small, recognizable number of categories, which might reflect different ways in which students engaged with the task.

Using the E value, we were able to find five different categories into which student sessions fell. We show below the E graphs for representative students in two of these categories, and give our interpretation of the problem-solving strategy used by the whole group of students. For these two categories, we have verified our interpretations by complementing the E value with further information from the log files.

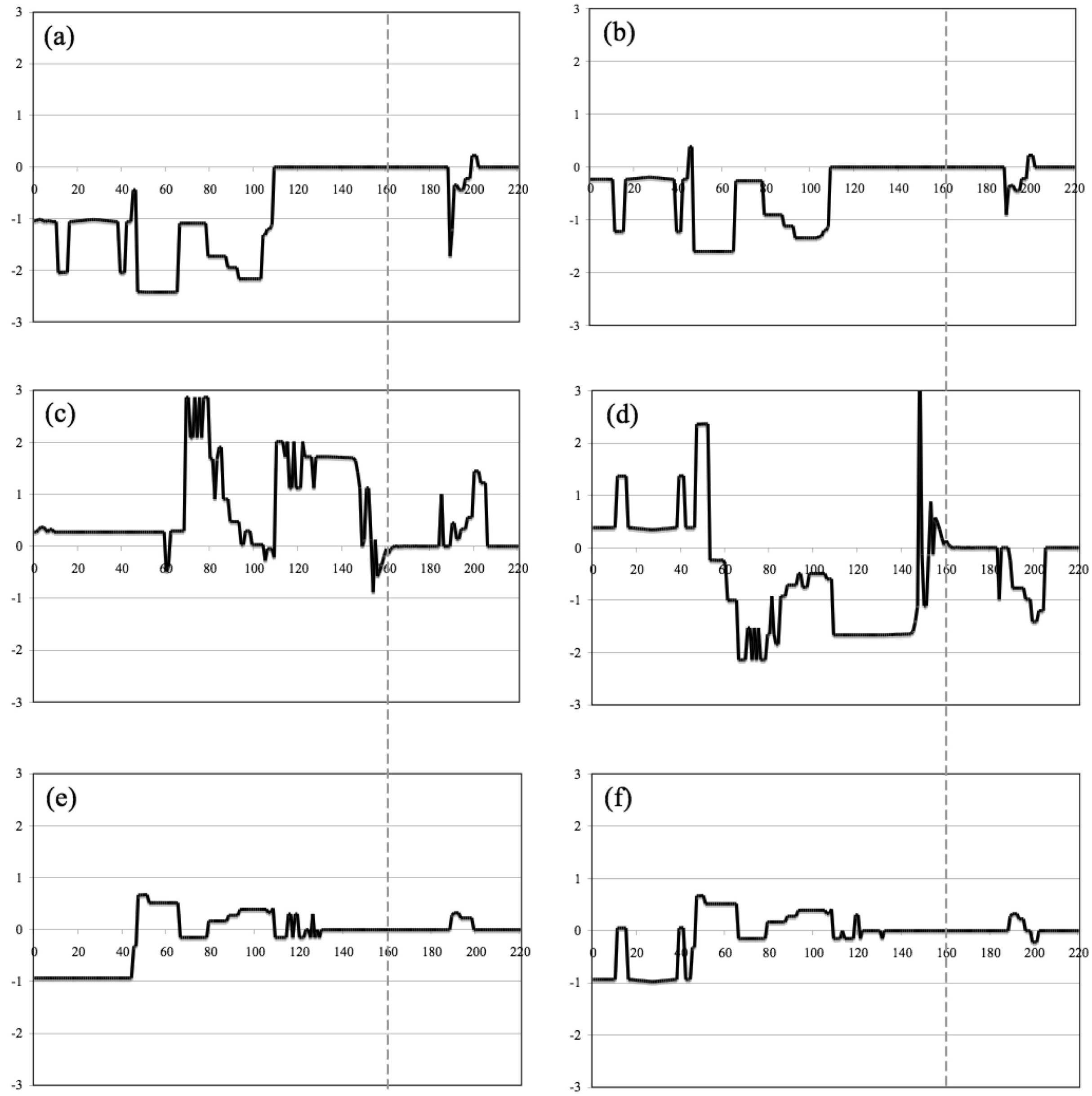

Fig. 1. Example of a single student session with each variable shown separately. The x-axis shows time, expressed as actions that changed an input variable. The y-axis is the logarithmic ratio of the student solution and the single best solution, normalised to the value of the solution. The overall session duration was 220 time intervals. The single best solution was derived at time interval 161, after which the student entered an exploration phase. Panels are: (a) Cost. (b) Mass. (c) Safety. (d) Force-Area. (e) Slenderness (Transverse). (f) Slenderness (Frontal).

4.1 Category 1: Student arrives at the correct solution, then stops

Students in Category 1 arrived at a solution, as shown by reaching an E value of less than Δ, and then terminated their session. A representative student session is shown in Fig. 3. After setting the problem physical data by step 16 (signified by the Cost error indicator reaching zero), this student then started moving toward the solution by adjusting the *Load* controller, reaching the solution at step 62. The student would have been optimizing the Factor of Safety (visible on the responder screen), and we can see in Fig. 3 that the magnitude of the logarithm of the error in the Factor of Safety is also moving toward 0 from step 16, and that the student over-shot the correct load around step 30.

We compared student sessions in this category with a session logged by an expert solving the same set task. While this student took 62 steps, the expert used 6 steps only, most of which were used to set the fixed parameters of the problem (size, shape, and material). Overall, within this group of students, the problem was solved, *i.e.* E is within Δ of 0, within a range of 16 to 554 steps (standard deviation 104).

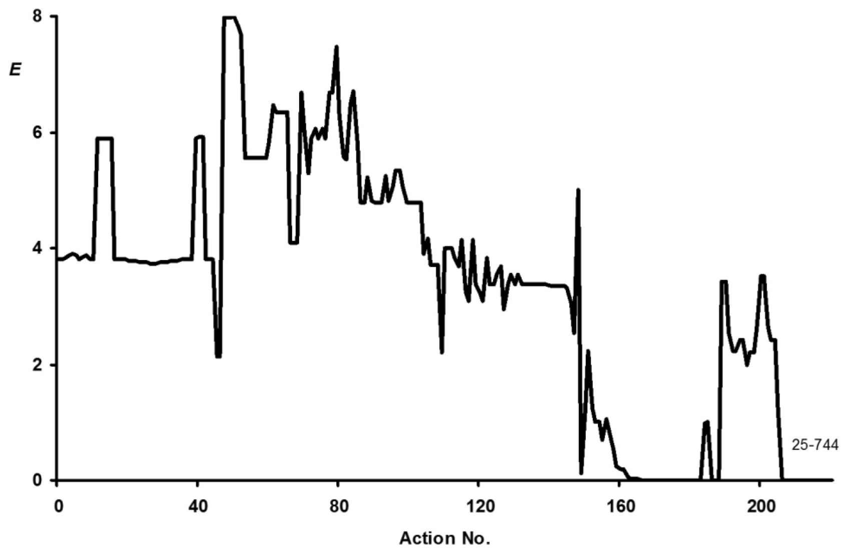

Fig. 2. Display of the same student session shown in Fig. 1, but showing the aggregate measure *E*, against time steps, instead of values of each of the output variables.

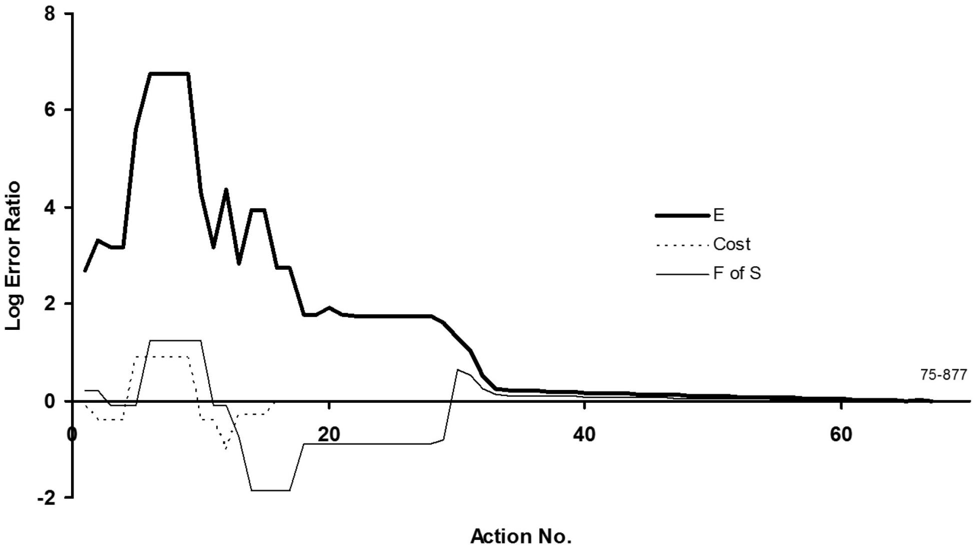

Fig. 3. Diagnostic parameters for a student in Category 1, reaching the correct solution after 62 actions.

Reasons for the large variation in the time to solution will be elaborated in the Discussion section.

4.2 Category 2: Student arrives at the correct solution, then explores

The SiMLED software has scope for creativity, and for exploring the effect of changing the values for different input variables. Several students in Category 2 arrived at the correct solution, as had the students in Category 1, again as shown by reaching an *E* value of 0. Unlike the students in Category 1, however, these students continued the session. A representative student session was shown in Fig. 2, and discussed in the previous section, where we described the usefulness of the metric *E*. While students in Category 1 were using SiMLED as a design tool to solve a problem, students in Category 2 were using SiMLED to explore the effect of changing inputs, as well as to solve a design problem.

4.3 Categories 3 and 4: Student solves the wrong problem

In Categories 3 and 4, the plot of *E* against time in the student sessions looked superficially similar the plots in Category 1 and Category 2: that is, the value of *E* tended to converge. However, the value to which *E* converged was not close to 0. Fig. 4 shows an example session for Category 3, where the time series for the student session flattens, and then stops.

As can be seen in Fig. 4, the last 40 steps of this

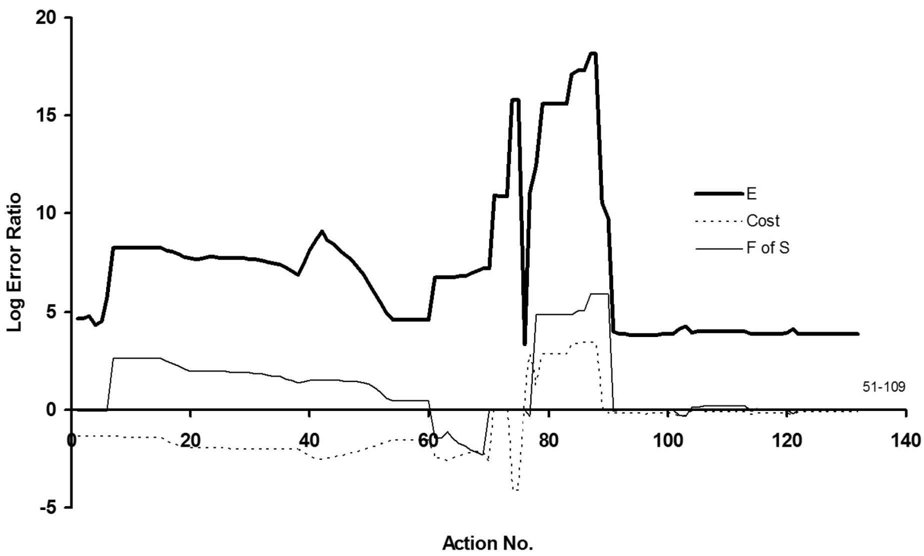

Fig. 4. Diagnostic parameters for a student in Category 3, reaching a wrong solution after 92 actions.

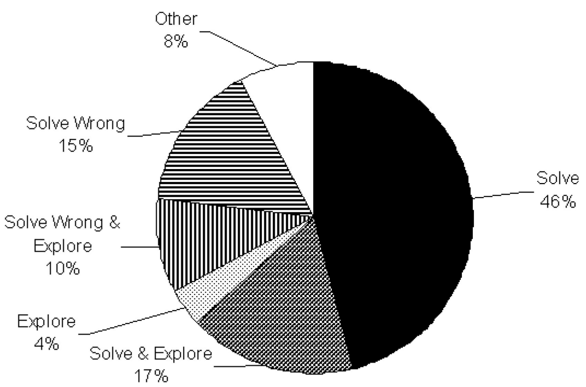

Fig. 5. Percentage of students falling into the various categories described in the text: Category 1 ("Solve"); Category 2 ("Solve and Explore"); Category 3 ("Solve Wrong"); Category 4 ("Solve Wrong and Explore"); and none of these categories ("Explore" and "Other").

student session show small changes in E. The E value hovers around $E = 4$, however, and not around the correct solution, where $E = 0$. The small changes in E towards the end of the session, seen in Fig. 4, like the small changes seen in Category 1 (Fig. 3), suggested to us that the student might have *thought* they were homing in on the correct solution. Further examination of the log files show that during these final steps of the solution, the error ratio for the Factor of Safety, the value the student was most likely tracking, indeed does hover near a value of 1 (shown by the log Error Ratio for the F of S in Fig. 4), as one would hope for when solving a structural problem. Having a Factor of Safety near 1, but an E value far from zero suggested to us that the student was solving *a* problem, but not the set problem. Further detailed examination of this log file confirmed this hypothesis. In fact, this student had altered the pre-set end constraints for the column, and then went on to solve the problem with incorrect end constraints. Several other students fell into this category, which included the selection of an incorrect cross sectional profile.

Category 4 bears the same relationship to Category 3 that Category 2 bears to Category 1: that is, in Category 3 the students solved the wrong problem, then stopped, while in Category 4 the students solved the wrong problem and then explored further.

4.4 *Category 5: Apparently not solving any problem*

In the last category, we have students whose log files show no signs of systematically working to solve *any* problem. That is, although the constraints were often set up at the beginning of the session, interrogation of the log files showed the effects of changing input variables were not then explored in a systematic fashion. We cannot rule out, however, the idea that these students were exploring the effects of changing input variables; we only know that they were not exploring them in the systematic way that we, as experts, would expect. Fewer than 10% of students studied fell into this category ("Other" in Fig. 5).

5. Discussion

While most studies in the emerging discipline of Learning Analytics focus on outcomes, that is on how much students learn, we have taken a process-oriented approach. Methodologies for estimating how much students learn exist (see, for example, Breslow [18]). However, these measurements are always proxies for what the student has learned.

For example, high scores on a test might signify successful learning, but might instead simply reflect good test-taking skills, or easy tests. Neither can the depth of learning be determined using standard measures. As Siemens has pointed out in a recent review [15], studies in Learning Analytics need to extend their approach to capture the complexity of the learning process.

In this study, we have taken a more process-oriented approach. We have the used log files for characterizing patterns of student learning where the students of mechanical engineering design were using simulation-based teaching educational software. In this environment, we ask the question "*How* do the students learn?", rather than the more subjective questions "*What?*" or "*How much?*" do they learn. To answer the question of how students learn in this environment, we use data obtained by directly tracking student actions while they are using this software. Although Clow has questioned the usefulness of usage tracking in Learning Analytics [11], his objections pertain to its limitations for outcome-focussed work, and do not apply to process-oriented studies such as ours.

Advantages of the process-oriented approach include:

- The ability to use direct data, not proxies.
- The potential for further calculation and informed manipulation to enhance the information obtained from the direct data.
- Feedback on the learning process to the educator, which in turn can lead to improved pedagogy.
- The potential for real-time intervention as students are using educational software.

5.1 Direct data

A clear advantage of tracking and using direct data about student usage is the objective nature of the data. A click of a button on the computer screen is a click of the button, and cannot be anything else. Our use of time series analysis ensures that the data are used directly, not just summaries of the data.

An advantage of using log files and plotting time series is that we do not overinterpret single numbers. As noted in Section 4.1, the number of steps students took to achieve a solution to this set problem varied from 16 to 554. If these numbers are treated superficially, then they would seem to say that there is a large variation in how quickly students were able to solve this problem. Buried within this range of 16 to 554 steps, however, when we look more closely, we find a number of different approaches. The SiMLED software has been designed intentionally to give students a variety of approaches, and to encourage them to explore.

For example, in the student session shown in Fig. 3, the student approaches the solution (i.e. approaches E = 0), relatively quickly between steps 28 and 33, and then much more slowly for another almost 30 steps thereafter. Examination of the log file shows that after the student set up the physical data by step 16, they began to change the applied load by using an 'inching' option that lowered the load by about 1% on each click, with each click counting as a separate action in the log file. This 'inching', gradually increased the Factor of Safety until step 28, when the student opted to use a 'thumbwheel' that lowered the applied load much more rapidly, i.e. using fewer steps. At step 33 this student reverted to the inching controller, slowing progress toward the solution. Other students used alternative input methods such as typing in trial values of the applied load, which generally reached the correct solution sooner. Although students using the inching option might superficially appear to have learned more slowly, this is an artefact of the flexibility built into the system.

Other students showed a similar fast-then-slow profile. We found from time series analysis of the log files that several of these students were seeking a Factor of Safety of up to eight decimal places, successively adjusting the sensitivity settings within SiMLED and incrementing the load by smaller and smaller amounts, leading to the apparently slow approach. Most students realized that the default setting of two decimal places would have been a more realistic stopping point.

Students were not instructed to solve the set problem as quickly as possible. Consequently, several students explored different materials and shapes for the column along the way to the solution, producing sudden swings in F of S, Cost and, consequently, E, thus simultaneously slowing their progress toward a solution and increasing their learning. Thus time series analysis can bring a richness to the analysis of students' learning that single numbers, and particularly aggregate numbers, overlook.

5.2 Further calculation

While it is common to talk about metrics in Learning Analytics and in Academic Analytics [11], these metrics are generally simply values that are measured directly, such as "Test results" or "Attendance". For the mechanical engineering design simulation program we were working with, we initially followed a similar approach, tracking the values of the separate output variables individually. However, we eventually found that combining the individual variables into a single metric gave a better intuitive idea of what was going on.

Developing a combined metric requires domain knowledge and thoughtfulness. In our case, for example, after some experimentation it was clear that we needed to measure an error, or distance from the correct output value, rather than the answer itself. We also used a logarithm scale to help minimize the differences between various variable values, which might have different scales. Similarly, a logarithmic scale helped equalize the differences between errors in different values.

In our experience, it has been important to separate the metric calculation from the logging of variable values. That is, we do not attempt to instrument the software to log the value for a combined metric, but only the direct variables. The combined metric is calculated during post-processing of the direct data. Separating the metric calculation from the data capture allows more flexibility than embedding calculated metrics. It allows the metric to be changed, refined, or added to as becomes appropriate. While there is some degree of problem-specificity in a metric, our recommendation is to keep the metric as general as possible, in the interests of reusability. Our formulation of E, for example, can generate distinct learning patterns for all of the SiMLED Columns learning tasks that we have set.

Two important consequences of using the combined metric E can be seen immediately. First, we can see the potential for automatically noting that a student has found the correct solution, simply by setting a flag in the software to note when E is less that the small value of Δ. This could facilitate data collection by the educator, and opens the door for real-time feedback to the student as they are actually using the software. Second, we can see the potential for automatically inferring student strategies from the convergence or fluctuations of E.

In our study, the ability to track single outputs (for example, Factor of Safety, or Cost), as well as the aggregate value E allowed us to understand what the students were doing. The aggregate metric would be generally useful for most problems, while tracking single output variables is usually a more problem-specific approach. Sometimes, the most power comes from looking simultaneously at two or three measures, even when one of them (for example, E in our study) is complex.

We would recommend that educational software be equipped with the capacity for logging actions, and with support for educators to develop more complex metrics on top of this. This layered approach, allows for flexibility. Although in a layered architecture, the combined variable, in our case the metric E, is calculated after the data is captured, there is no need to write the whole log file before engaging in post-calculation. The metric E could be calculated quickly enough to give the impression of real-time to the student where needed.

We suggest that the practice of combining the values of output variables could be useful even in less technical domains. For example, it might be useful to combine variables such as "time to finish test", "test score", and "laboratory attendance". Bienkowski et al. [1] have even suggested that there might be merit in bringing together administrative and classroom-level data.

5.3 Feedback to the educator

Our work has shown that it is possible to detect various patterns of student behaviour using data capture, time series, and calculating complex metrics. We have been able to use this information to gain insight into the student learning process. User behavior modelling can fruitfully ask the question: "What do patterns of student behavior mean for their learning?" [1].

Part of the beauty of the process-oriented approach is that we are able to detect patterns of learning that we had not anticipated. In the current study, we were able to detect several students who set out to solve the wrong problem, and often solved that problem correctly. Our unexpected identification of significant Categories 3 and 4 (together, 25% of the cohort), students who solved the wrong problem, is an example of how metrics can be used by the educator to (a) better understand student learning and (b) improve pedagogy. Because we were looking at the process, we were able to note students who had not set the constraints as stated in the specification. Had we been looking for learning outcomes, rather than at the learning process, we would probably have considered students in these categories as having poor learning, because they did not get the correct answer. In fact, from our study we can see that these students actually had been able to solve a problem, but had started with a misunderstanding of the problem. Looking at the students in these categories further, we also noted where the educator might have been able to further clarify the problem at the start.

We detected yet another pattern in the log files. Students in Category 5 did not set up the problem constraints, and changed variable values in a disorganized fashion. These students did not seem to know how to go about solving the problem. Again, having this feedback, the educator might reflect on whether more instruction on approaches to problem solving might be useful incorporated into the classroom.

In a previous study [8], we have noted that student goals in using educational software are often different from the educators' goals, and lead them to use software in ways not initially envisioned by the educator. In some cases, these unintended patterns of students learning can be quite productive. Detecting these patterns, and bringing them to the attention of the educator, means they can be incorporated into the classroom pedagogy.

Our findings of distinct categories also pointed out how monitoring log files using metrics can help the educator clarify tasks for the current and future cohorts of students.

5.4 Potential for real-time intervention

Another advantage of taking the time-series approach to analytics is the potential for real-time intervention. For example, when students have been solving the wrong problem (Categories 3 and 4), that erroneous track would show up in the log of actions as a slow approach to a non-zero value of E. The log would also show an approach to a Factor of Safety of 1. For active intervention, the slowly decreasing E value could trigger a check for the E value being approached, and if it is not zero, then a check on the progress of the Factor of Safety. Together these two situations could flag a high likelihood of solving the wrong problem, and importantly could trigger a message sent to the screen, such as a pop-up window, warning the student that they might be looking at a problem other than the set problem. Of course, the student who prefers to explore unaided should have the option to disengage online guidance.

The layered approach mentioned above, where basic actions are logged and complex metrics are built on top of the logged actions, would help increase the flexibility of this kind of intervention. Interventions could be designed to be suitable for different situations and different problems. We note that even though some of the more complex metrics need to be calculated from the direct student actions, and therefore after these actions are recorded, and the triggers would be detected after the variables are constructed, the lag would be on the order of microseconds, so in effect the student would be seeing "real-time" intervention.

5.5 Limitations of this study

Certain conditions must be met before an educational software metric can be used effectively to follow student activity and monitor progress over the course of a learning session. Most importantly, the relevant outputs need to either be numeric, as here, or categorical values that can be assigned relevant numeric values for subsequent analysis. The vast majority of engineering education problems meet this condition.

In principle, the technique proposed here is applicable for any type of problem where there is a definable criterion for a 'best' or 'better' solution, under any set of constraints. While we have based our experiments on a task where there is a single, known best solution, this is not an absolute requirement. Tasks with multiple best solutions are also suitable for this approach, although eliciting fine details of the learning progress might require a slightly more complex analysis. The technique can be effective for any multi-variable or multi-solution problem, as long as reasonably continuous numerical variables are part of the solution.

6. Conclusion

In our study, we have found that we can usefully track student learning using software through recording their actions in log files. Analyzing time series data in the SiMLED environment, as students solved a mechanical engineering design problem, we detected different patterns of learning and gained insight into the student learning process. A particularly useful metric was developed by combining several direct outputs. We suggest that usage tracking of this sort can support real-time feedback to students on their progress and can be generalizable to other technology-based learning systems.

This project has been carried out with the approval of the Human Research Ethics Committee of the Faculty of Engineering at The University of Melbourne.

References

1. M. Bienkowski, M. Feng and B. Means, U.S. Department of Education, Office of Educational Technology, Enhanced Teaching and Learning through Educational Data Mining and Learning Analytics, https://tech.ed.gov/wp-content/uploads/2014/03/edm-la-brief.pdf, Accessed Nov 19, 2015.
2. M. D. Byrne, R. Catrambone and J. T. Stasko, Evaluating Animations as Student Aids in Learning Computer Algorithms, *Computers & Education*, **33**(4), 1999, pp. 253–278.
3. J. Roschelle, N. Schechtman, D. Tatar, S. Hegedus, B. Hopkins, S. Empsom, J. Knudson and L. Gallagher, Integration of technology, curriculum, and professional development for advancing middle school mathematics: Three large-scale studies, *American Educational Research Journal*, **47**(4), 2010, pp. 833–878.
4. Z. Hussein and L. Stern, The Effect of Goals on Use of Educational Software, *Proceedings of World Conference on Educational Multimedia, Hypermedia and Telecommunications (ED-MEDIA 2008)*, Vienna, Austria, 2008, pp. 5640–5645.
5. G. E. Kennedy and T. S. Judd, Making sense of audit trail data, *Australasian Journal of Educational Technology*, **20**(1), 2004, pp. 128–132.
6. L. Stern and K. Lam, A Framework for Evaluating Multimedia Software: Modeling Student Learning Strategies, *Proceedings of World Conference on Educational Multimedia, Hypermedia and Telecommunications (ED-MEDIA 2007), Vancouver*, Canada, 2007, pp. 3301–3309.
7. K. R. Koedinger, E. Brunskill, R. S. J. Baker, E. A. McLaughlin and J. Stamper, New potentials for data-driven intelligent tutoring system development and optimization, *AI Magazine*, **34**(3), 2013, pp. 27–41.
8. C. Romero, Educational Data Mining: A Review of the State-of-the-Art, *Transactions on Systems, Man, and Cybernetics – Part C: Applications and Reviews*, **40**(6), 2010, pp. 601–608.
9. A. van Barneveld, K. E. Arnold and J. P. Campbell,

Analytics in Higher Education: Establishing a Common Language, *Educause Learning Initiative 2012*, http://net.educause.edu/ir/library/pdf/ELI3026.pdf , Accessed January 18, 2013.
10. J. P. Campbell, P. B. DeBlois and D. G. Oblinger, Academic Analytics A New Tool for a New Era, *Educause Review*, **42**(4), 2007, pp. 41–57.
11. D. Clow, An Overview of Learning Analytics, *Teaching in Higher Education*, **18**(6), 2013, pp. 683–695.
12. R. Ferguson, The State of Learning Analytics in 2012: A Review and Future Challenges, *Technical Report KMI-12-01*, Knowledge Media Institute, Open University, United Kingdom, http://kmi.open.ac.uk/publications/techreport/kmi-12-01, Accessed 10 Dec 2015.
13. D. M. Norris, 7 Things You Should Know about First-Generation Learning Analytics, *Educause Learning Initiative 2011*, http://www.educause.edu/library/resources/7-things-you-should-know-about-first-generation-learning-analytics, Accessed 17 February 2014.
14. P. Long and G. Siemens, Penetrating the Fog: Analytics in Learning and Education, *Educause Review*, **46**(5), 2011.
15. G. Siemens, Learning Analytics: The Emergence of a Discipline, *American Behavioral Scientist*, **57**(10), 2014, pp. 1380–1400.
16. A. E. Samuel and J. G. Weir, *Introduction to Mechanical Engineering Design*, Butterworth-Heinemann, Cambridge, 1999.
17. J. G. Weir, C. R. Burvill, A. E. Samuel, Simulation modules for learning engineering design (SiMLED), *12th International Conference on Engineering Design (ICED 99)*, Munich, Germany, 1999, pp. 899–904.
18. L. Breslow, Methods for Measuring Teaching and Learning, *Teaching and Learning Laboratory*, MIT, 2007 https://tll.mit.edu/sites/default/files/guidelines/a-e-tools-methods-of-measuring-learning-outcomes-grid-2.pdf, Accessed 7 December 2015.
19. S. Schmid, B. Hamrock and B. Jacobson, Fundamentals of machine elements, CRC Press, Boca Raton, USA, 2014.

Appendix: The SiMLED Learning Platform

In this study we used the software package SiMLED (Simulation Modules for Learning in Engineering Design) as a concrete example of the kind of software that can be used by students to learn about complex problems. SiMLED was developed in the Department of Mechanical Engineering at The University of Melbourne to facilitate student learning of mechanical engineering design [16, 17]. SiMLED was developed as a tool to solve directed tasks in mechanical engineering design and for open-ended exploration.

Simulations in the SiMLED modules allow students to change design parameters, such as material of manufacture and structural dimensions, and then allows them to observe the effect of these design changes on the structure and functionality of the component they are designing. The process is similar to experimenting in a physical design laboratory, but allows experimentation with a much wider range of dimensions and materials than would ever be available in a physical environment. The essential characteristic of the object being modelled is that its various features are defined parametrically, and that the student user can modify the defining parameters interactively.

In this work, we have concentrated on the SiMLED learning module *Columns*, where buckling and crushing modes of failure can be investigated. When SiMLED is first opened, the student is presented with a choice of input design parameters, controlled by the tabs, shown on the right hand panel in Fig. A and three different representations for observing the effect of changing a parameter in the left hand panel of Fig. A.

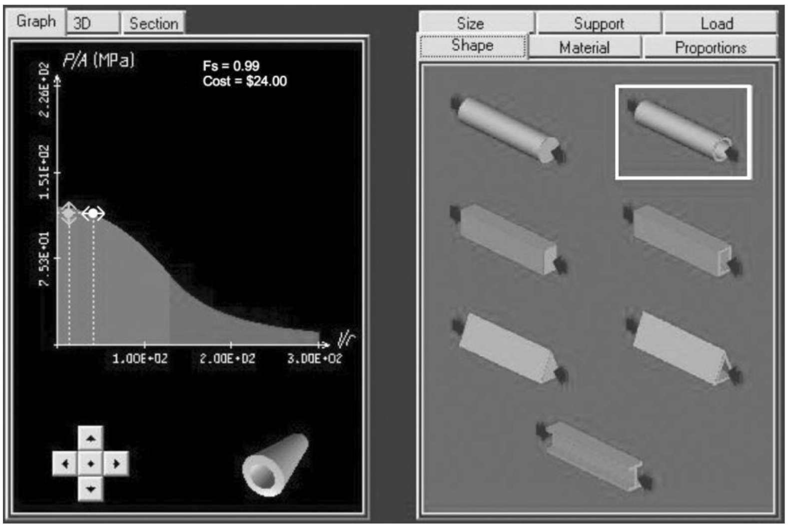

Fig. A. Start up SiMLED user interface, showing the Euler-Johnson column design curve, the Factor of Safety (Fs) and the Cost in the Graph responder window on the left and alternative generic shapes on the right, with the hollow cylinder (represented visually) chosen. There are three alternative tabbed responder (viewing) windows on the left and six alternative controller (input) windows on the right.

Students can input design parameters by accessing one of the six "controller window" categories: Shape, Material, Proportions, Size, Support, and Load (Fig. B). Each category will have one or more input design parameters. For example, the Shape window allows students to change the generic sectional shape of the object, and the Size window allows students to modify both the length and height of the object.

When a student changes the input design variables, the dependent variables, or performance variables change in response. There are six dependent variables for the Columns module: *Cost, Mass, Safety, SlendX* (slenderness for buckling in the X (*Transverse*) direction), *SlendY* (slenderness for buckling in the Y (*Frontal*) direction), and *PonA* (axial load per cross-sectional area).

In the computer screen responder window, the changes are visualized, as if the student were working with physical forms. The visible form of the column and the values for the dependent parameters are shown in the responder window (Fig. C). The graph view is particularly useful in helping a student find an acceptable solution.

The kind of information available from the graph view of the responder window is illustrated in Fig. D, where two instances of the responder window are shown. The graph shows the Euler-Johnson curve and

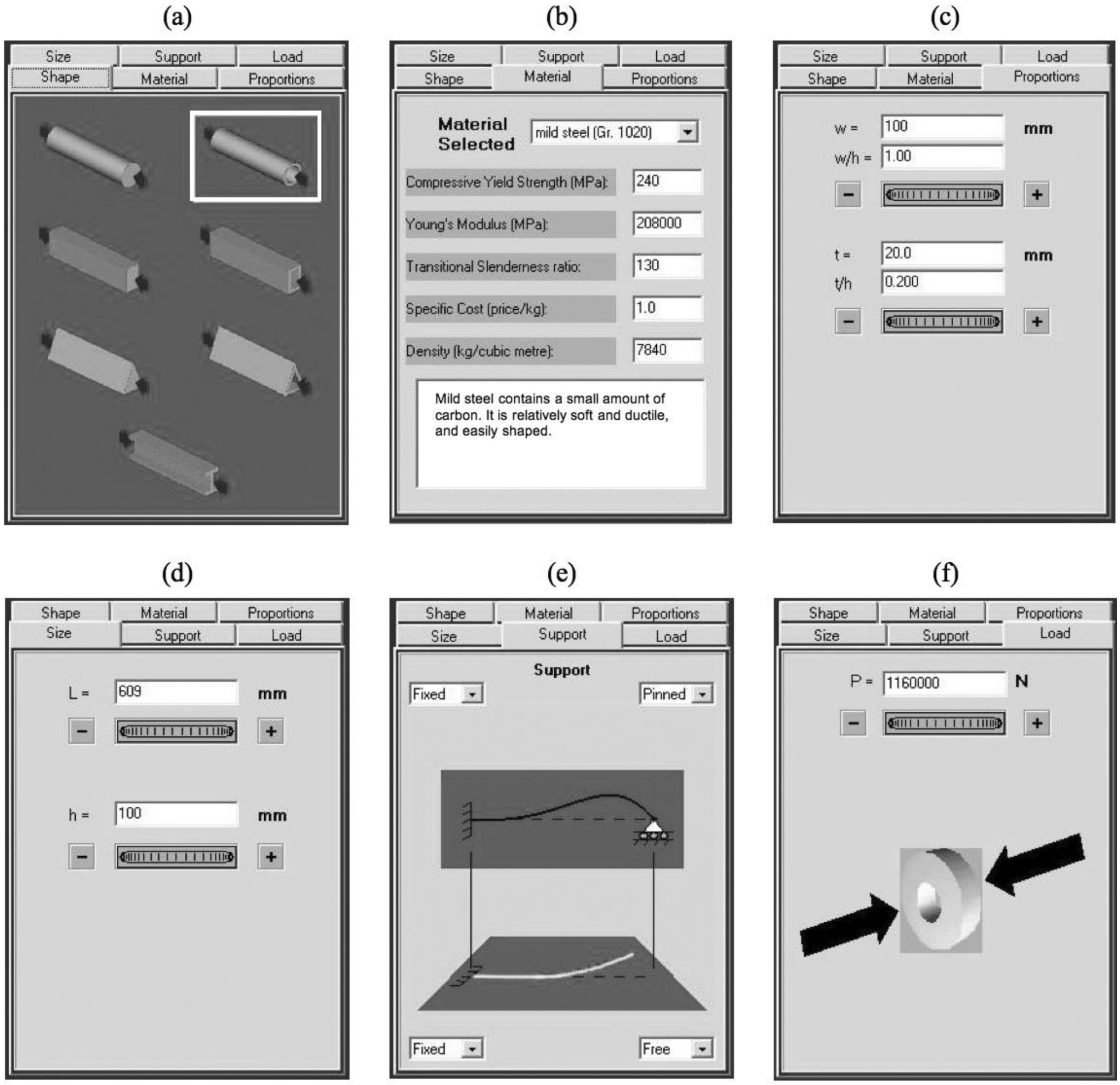

Fig. B. Controller windows. Sample SiMLED controller windows used for inputting the numerous design variables in the Columns module. Controller variables: (a) Shape, (b) Material, (c) Proportions, (d) Size, (e) Support, (f) Load. The category of variable is chosen via a tab at the top of the window, while specific values and selections for individual design parameters can be selected via drop-down menus, accessed by clicking on an icon, typed into the text boxes, altered by thumbwheels or incremented by buttons.

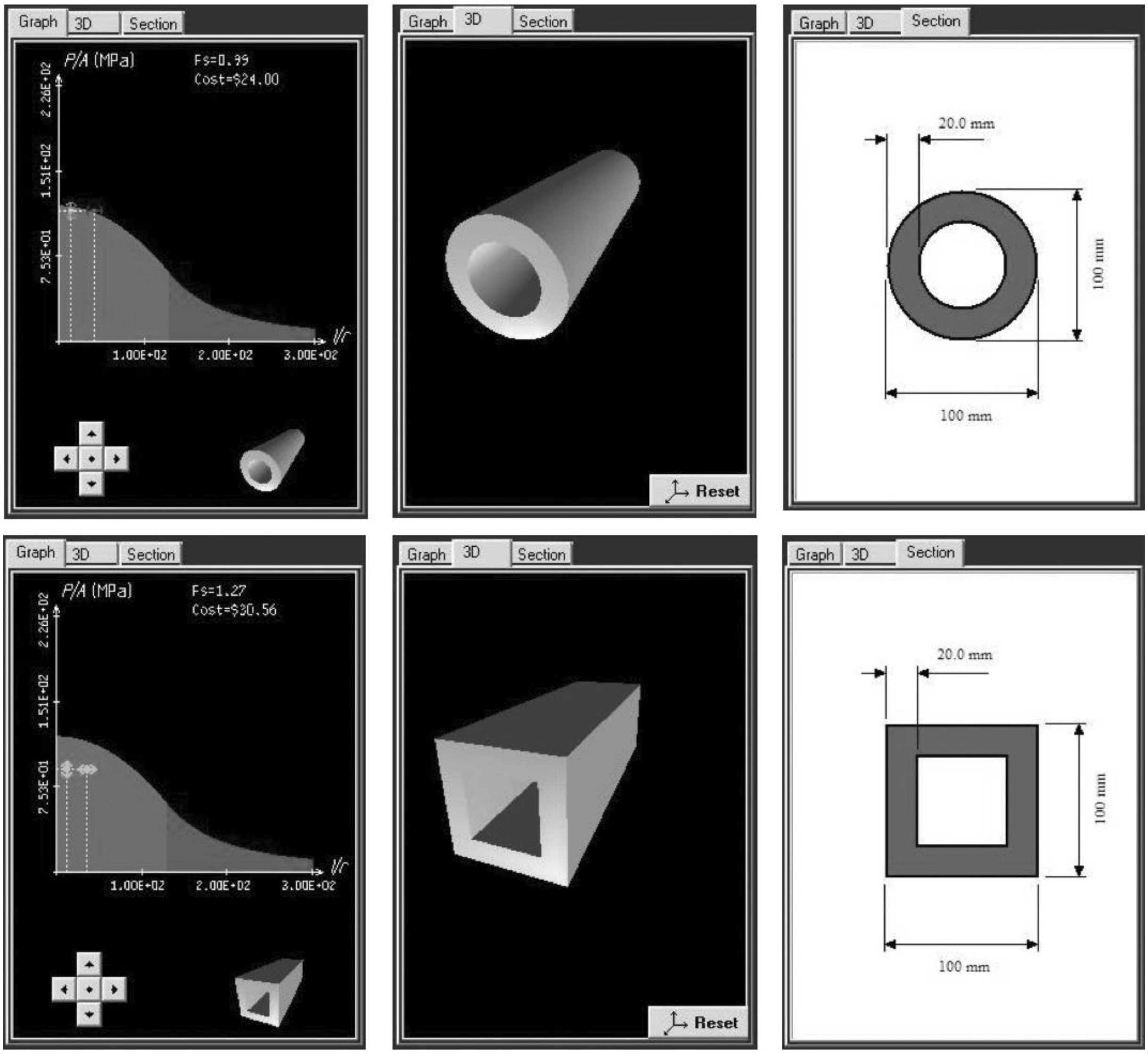

Fig. C. Two sets of alternative SiMLED Responder windows, as shown in the tabs at the top of the windows, the views are: Graph, 3-Dimensional, and Cross-section. The top three panels show the three views for a hollow circular cylinder (pipe). The bottom three panels show the three views for a rectangular hollow section (hollow square).

Factor of Safety for the column, given the design variables that the student has selected [17]. The shape of the curve will change, for example with material selection, reflecting the underlying column design theory [19]. The arrows in the lower right corner of the graph view enable the student to scale the axes of the graph to better observe the consequence of parameter changes. The lower right corner shows a small icon of the Shape chosen. This ensures that the student is under no misapprehension about the shape under consideration and saves the student from continually flipping between graph and 3D or section views. The graph view can be helpful for students searching for design parameters within a reasonable factor of safety, *i.e.* an acceptable solution. Fig. D also highlights the importance of the target value cross-hairs associated with the Euler-Johnson curve. Within the theory of column design, an unsafe design can occur in one of two orthogonal directions, and the "directional target" cross-hairs highlights this important fact, *i.e.* potentially "safe" in one direction and "unsafe" in the other.

The design safety factor F_s (*F of S* in Figs 3 & 4)) corresponds to the mode of failure that is most likely to cause a potentially unsafe design scenario (a loss of structural integrity). The cross-hairs in Fig. D represent F_s independently in the plane of the chart, *i.e.* the x- (l/r) and y- (P/A) directions, to encompass non-axisymmetric design scenario. The directional target cross-hairs change position dynamically as the independent variables are changed.

For structural integrity, the entire target must be within a shaded area of the Euler-Johnson curve (Fig. D, top, with $F_s > 1$). The confidence the student designer can assume for the interim column is reinforced by the

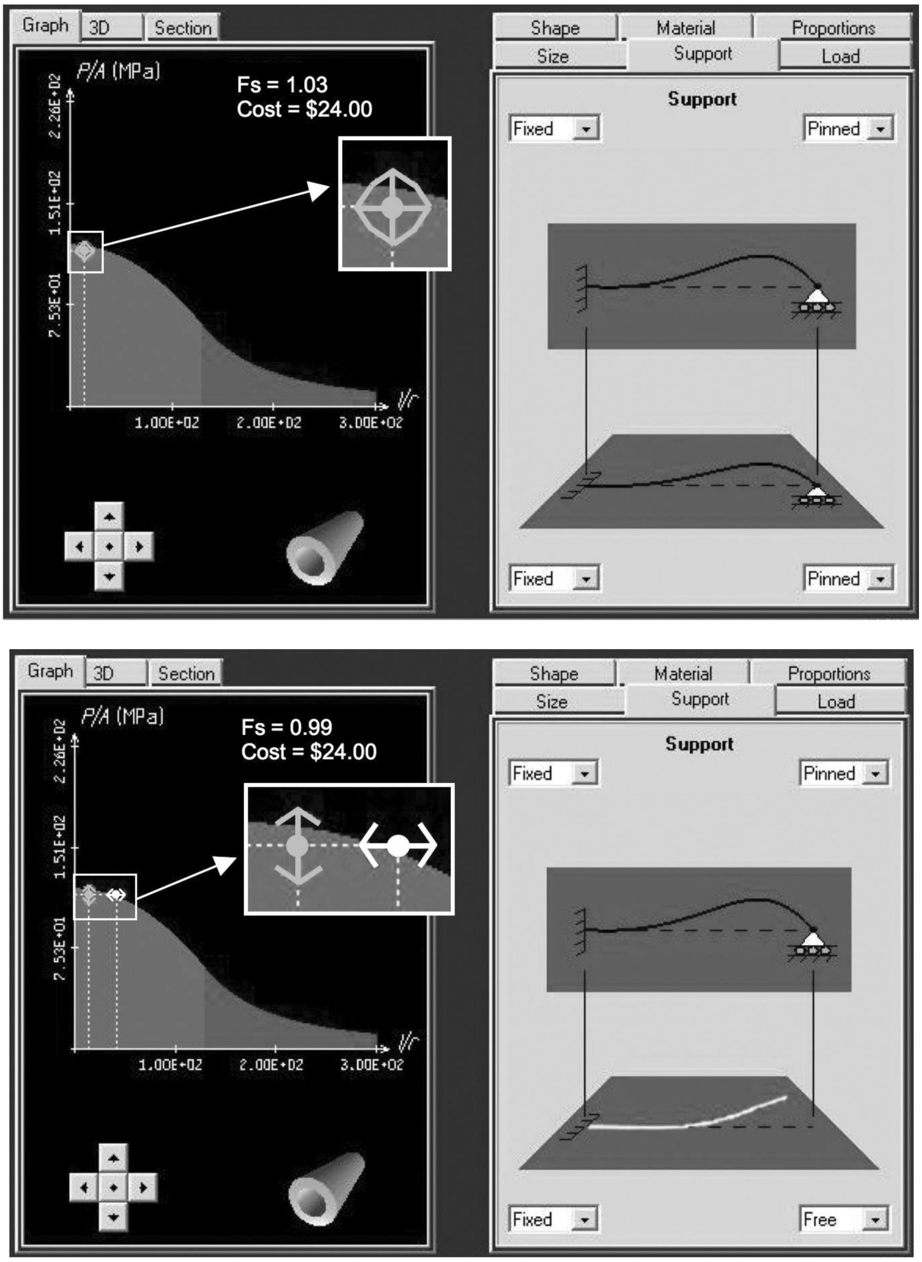

Fig. D. SiMLED user interface, showing the Euler-Johnson column design curve in the Graph responder window on the left and the Support controller window on the right.

In the top panels, support is the same in both directions (overlapping grey icons both show Fixed at the left end and Pinned at the right end). According to the diamond shaped ''directional target'' cross-hairs, the associated column design is safe.

In the bottom panels, the support has been made asymmetrical. The directional target now separates into two two-headed arrows, losing its prior appearance. The upper Fixed-Pinned direction, associated with the grey vertical two-headed arrow icon in the graph responder window, is identified as being safe. The lower Fixed-Free direction is identified as being unsafe, with both the column icon and the horizontal two-headed arrow (shown in red on the student screen), to underscore the problem with this unsafe design scenario, in particular the most likely direction of failure.

numerical value of the design safety factor, F_s shown in the upper right corner of the graph window. As the target moves further from the Euler-Johnson curve towards the origin of the graph, F_s increases in value and the interim design becomes safer. Of course, conflicting issues become apparent with increasing F_s, in particular, increasing cost and associated inefficiency. The software has been described in more detail elsewhere [17].

The facility to record student actions has been incorporated into the software as described previously [4]. An XML log file records significant student actions, and includes: (1) the name of the action, *e.g.* ''change responder window'', ''change material''; (2) the time of the action; and (3) any parameters relevant to that particular action, *e.g.* the material chosen, length, *etc.* Values of both the independent design variables and the dependent performance variables were recorded in a log file.

Linda Stern is a Senior Lecturer in the Department of Computing and Information Systems at The University of Melbourne (Australia). She joined the department in 1991, after several years of research in Molecular Biology and Clinical Microbiology. She has combined her interests in biology and computing and carries out both teaching and research in the area of Computational Genomics. She has been project leader for development of the educational teaching tool Algorithms in Action, and has published widely in the field of Computer Science Education.

Colin Burvill is a Senior Lecturer in the Department of Mechanical Engineering at The University of Melbourne (Australia). He joined the Department after an early career as a practicing project and design engineer. Dr Burvill is the Department's design Discipline Leader and is the coordinator and lecturer of all mechanical engineering design subjects and final year (capstone) project subjects, which includes administration and supervision of up to 70 projects annually, many with industry collaborators. He also provides project experiences to students completing a Master of Entrepreneurship degree within the University. Dr. Burvill's research interests include issues associated with ''design for manufacture and innovation'', encompassing aspects ranging from human cognition and creativity to mechanical systems design, biomechanical design and environmental design. He has successfully integrated research outcomes with company needs on a series of post-graduate and post-doctoral research and development projects.

John Weir is a consulting mechanical engineer with a focus on renewable energy systems, and a long commitment to engineering design education. His combined industry and academic career has been as joint design practitioner and educator. Prior to returning to full-time consulting practise, he lectured on all aspects of design to students in mechanical, manufacturing, civil, and environmental engineering, and was actively engaged in the development of a programme of industry/university student project teams. He conceived and produced the first implementations of SiMLED software for interactive design and modelling of basic structural components. He has co-authored an international textbook (Samuel and Weir, Introduction to Engineering Design, 1999). His industrial consulting experience includes design roles in mechanical equipment, chemical process plant, and pressure vessels, and the provision of advice to industry clients and legal proceedings in relation to wind turbines, solar thermal plant, lifting equipment, manufacturing processes, machine failures, and intellectual property.

Bruce Field is a consulting mechanical and forensic engineer who advises Australian industry and legal representatives on aspects of machine design, manufacturing, innovation and diagnostic problem-solving. He was formerly an Associate Professor in mechanical engineering at Monash University (Australia), with principal responsibilities in undergraduate professional education, including engineering design, engineering drawing, and innovation processes, and has published widely on these topics. He was also responsible for course coordination and reform in the introductory undergraduate engineering year at the University of Melbourne, and maintains an honorary position at that university, with co-supervision of undergraduate and postgraduate student projects.