

An electronic tutor for mechanical engineering

Bruce Field

Monash University, Clayton, Australia
Bruce.field@eng.monash.edu.au

John Weir

University of Melbourne, Melbourne, Australia
jweir@mame.mu.oz.au

Colin Burvill

University of Melbourne, Melbourne, Australia
colb@mame.mu.oz.au

***Abstract:** For a variety of safety and economic reasons, engineering undergraduates in Australia have experienced diminishing access to the real hardware that is typically the embodiment of their theoretical studies. This trend will delay the development of practical competence, decrease the ability to model and design, and suppress motivation in all but the most imaginative students. The authors have attempted to address this concern by creating a software tool that contains both photographic images of real machinery, and sets of modelling 'tools'. Academics can use the software to set tutorial tasks, and incorporate feedback comments for a range of student responses. An evaluation of the software demonstrated that students who had solved modelling problems with the aid of the electronic tutor performed significantly better in formal examinations with similar modelling tasks.*

***Keywords:** Modelling, structural distillation, CAL, tutoring*

Introduction

Among the multitude of difficulties facing 21st century engineering education are two that are growing more critical as time progresses: (a) pressures of reduced government funding have increased student/staff ratios and reduced opportunities for personalised tutoring, and (b) those reductions in funding, along with increased concerns about liability and safety have limited the opportunities for hands-on or otherwise realistic experiences for undergraduate engineers. When coupled with the shift toward student-centred learning (and the desire to match learning opportunities to individual needs) we find that those difficulties lead to students' perceptions of a widening gulf between engineering practice and engineering education, with increasingly scarce opportunities for connecting individuals with their future profession.

Those funding reductions are being addressed (to varying degrees of success) by a need for universities to earn income from other activities – mainly through research grants. Research success is consequently a desirable attribute for modern academics, while ongoing liaisons within the profession are less well regarded. Time, modest practical experience and funding

limitations also create a gulf between many engineers in academia and industry, and this has an impact on the programs of teaching offered in engineering courses.

The authors have observed that the majority of engineering sciences taught at their universities are bereft of the artefacts associated with the science, and in some instances, are bereft of realistic representations (e.g., photographs, videoclips) of those artefacts. Yet we have observed that students are highly motivated by the existence of realistic (“practical”) examples of the theory: it appears that an appreciation of abstractions requires some time to mature, perhaps even well after graduation for some. The experienced teacher-researcher already possesses this ability to abstract from reality, and often teaches from the abstraction, rather from the reality. This approach can lead to student dissatisfaction and their evaluation comments that a study unit is “too theoretical”.

The act of constructing an abstraction from reality is called modelling. Models may be mathematical/algebraic, physical, graphical or symbolic, or some combinations of these. When the models contain elements that are the basic building blocks of the engineering discipline, we call the modelling process that of ‘structural distillation’ (Samuel and Weir, 1999). The usual step following such a structural distillation is to find and include the numerical data that is specific to the discipline, thereby allowing the engineer to make specific predictions from the model.

It is evident that a practicing engineer should be able to form proper and correct models, uncover the data for their particular problem, and then ‘solve’ the problem to meet the final need (Samuel and Weir, 1991). Yet there is little evidence that undergraduate engineers are schooled in the art of structural. The gap in this ability became apparent to the authors, whose specialist teaching area is in mechanical design. In open-ended design problems the student may progress in either of two directions: (a) from a concept toward determining if that concept will work, or (b) from a calculated (numerical/graphical) descriptor of the requirements toward the physical embodiment that would achieve the requirements. In both instances the student needs to formulate a model that bridges the two elements, and this is often the most inadequately-performed task in an undergraduate design (Ferguson, 1992).

Most of the conceptual modelling tasks (structural distillations) needed in undergraduate problems are very basic, and require only a few minutes of effort (from a capable student). It is impractical to personally tutor large groups of students in this undertaking. With this restriction in mind, coupled with the concern for the reduction in ‘practicality’ in many undergraduate engineering courses, the authors conceived the potential for a computer-based tutor that could: (a) contain realistic representations of engineering artefacts (perhaps animated), (b) offer a method of allowing students to formulate structural distillations of those artefacts, and (c) implement a technique for correcting errors on an individualised basis.

Support for the development of such software was forthcoming in 2001 from a joint funding scheme created by the authors’ universities. The basic version of the electronic tutor was written during that year, and is called MOMUS Tutor (Monash-Melbourne Universities’ Structural Tutor). (Momus is also a word for a fault-finder, or persistent critic, derived from the Greek god of ridicule.)

This paper describes the underlying educational philosophies and program structure of the MOMUS Tutor, and reports the educational outcomes from its first year of use.

Principles of MOMUS Tutor

It was intended that the proposed electronic tutor should be available for structural distillations in any of the engineering sciences in a mechanical engineering course, and it was envisaged that, since the fundamental issues being addressed by the tutor were likely to be common across several disciplines (and outside of engineering), it might be possible to construct a fairly general tool. Since the engineering sciences tend to work in relative isolation, and students see this separation in their timetables, it was desirable to compartmentalise the modelling activities for each science. It was also recognised that the issues of modelling within one sub-discipline need to be tested over several different circumstances to ensure that the principles have been properly learned. These aspects led the authors to believe that it should be possible to construct a tutoring program with generalised capabilities, that could be customised for separate artefacts, sciences, and disciplines.

The requirements suggested a ‘grid’ of tutorial tasks, with artefacts and sciences forming the two axes, and one or more tasks made available in each of the grid elements. Figure 1 is the problem grid for MOMUS Tutor (February 2003 version). The intention was that problems could be generated in each of the grid elements, and students might select individual problems, sets of problems from an engineering science (a column), or the issues associated with an artefact (a row). The authors were especially interested in the formulation of problems along a row, since this should illustrate the integration of several engineering sciences within the design of an artefact (one of the primary purposes of an engineering design unit).

The artefacts themselves were intended to be realistic representations, so were likely to be information-rich (such as photographs, videotape, or rendered 3-D models), causing the artefact representations to occupy large volumes of electronic memory. To minimise the difficulty in manipulating such large amounts of memory, it was also desirable that the artefacts be suitable for modelling tasks in several sciences. At present, it is expected that the final version of the MOMUS Tutor with eight different machines will fit onto a conventional CD ROM, or be available through campus networks, although it may not be suitable for dial-in modem access.

MOMUS Tutor was coded in Macromedia’s Director, a common base for educational software. It allows for the simple creation of a stand-alone ‘projector’ for distribution, and compressed versions to be played by Shockwave (free downloadable software from Macromedia). Figure 1 shows the basic program structure. The starting ‘movie’ allows students to jump to an introduction, describing the purpose of the software and how to use it, or go directly to the problem movie. From the contents frame (Figure 2) one or more problems can be selected, then attempted in sequence. The hardware is represented in separate ‘frames’ of the movie, and the separate sciences are represented by independent ‘casts’ (Rosenzweig, 2000) of icons.

A problem is constructed from the combination of an artefact with the contents of a science cast, plus the specific text that is authored for each problem.

The potential to animate machines (in order to gain insight into their functionality) and the need to isolate portions of a machine in some modelling processes led to the decision to ‘construct’ the artefacts from separate images, ‘assembled’ together to show the whole machine. Figure 3 is the screen image of the first problem in the grid shown in Figure 2.

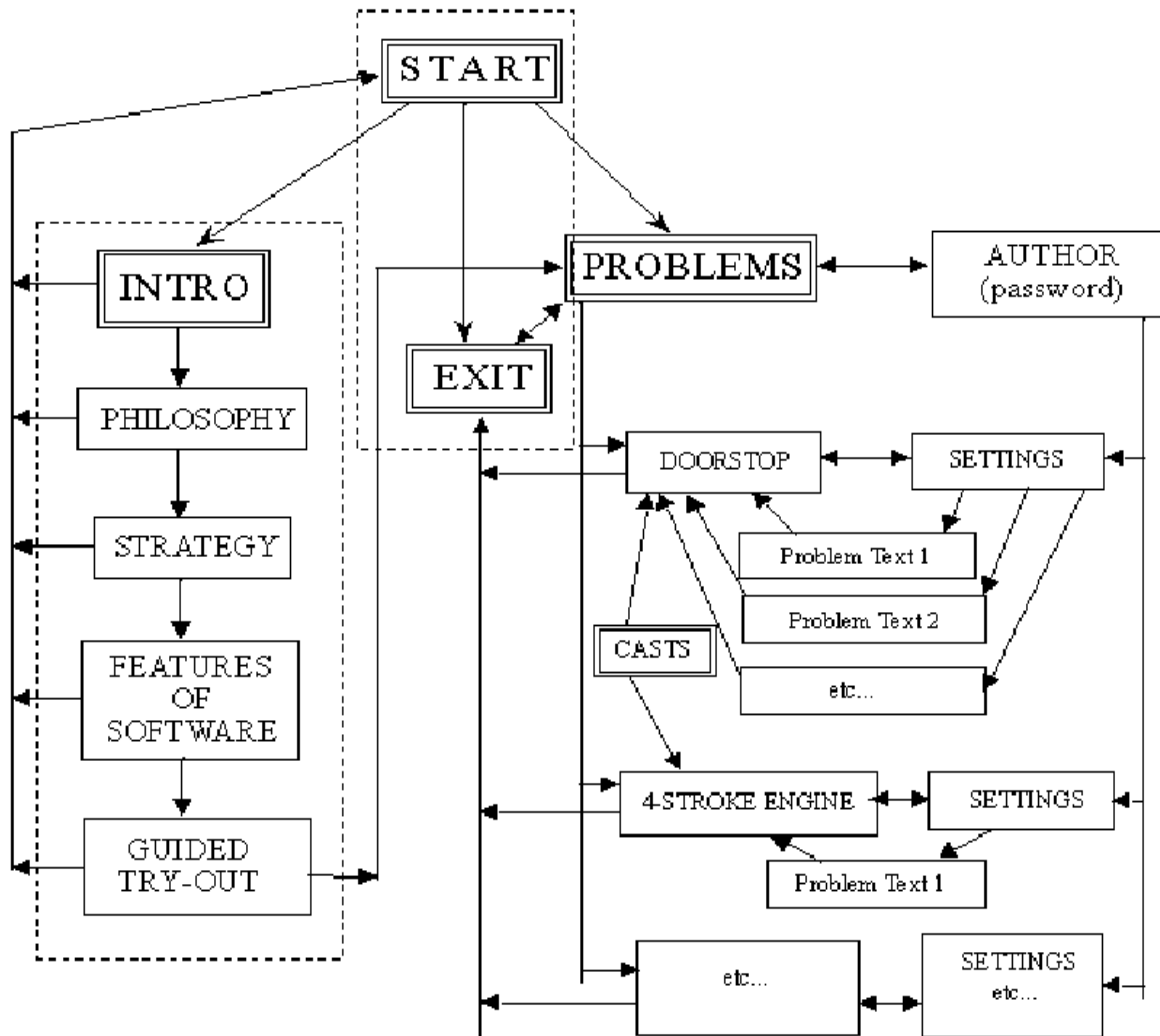


Figure 1: Structure of the MOMUS Tutor

CONTENTS

Choose the problems to be attempted in any of three ways:

- Select all problems of the same TYPE from the top row of type names.
- Select all problems for the same MACHINES from the column of machine names.
- Select individual problems from the highlighted alternatives.

The selection will appear in the list.

	STATICS	SOLIDS	DYNAMICS	HEAT	FLUIDS	CONTROL
DOORSTOP						
4-STROKE ENGINE						
FOOTPUMP						
REFRIGERATOR						
GAS TURBINE						
GOVERNOR						
FAN						
SUSPENSION						

Cancel
QUIT
BEGIN

Figure 2: Contents page of MOMUS Tutor with several problems available

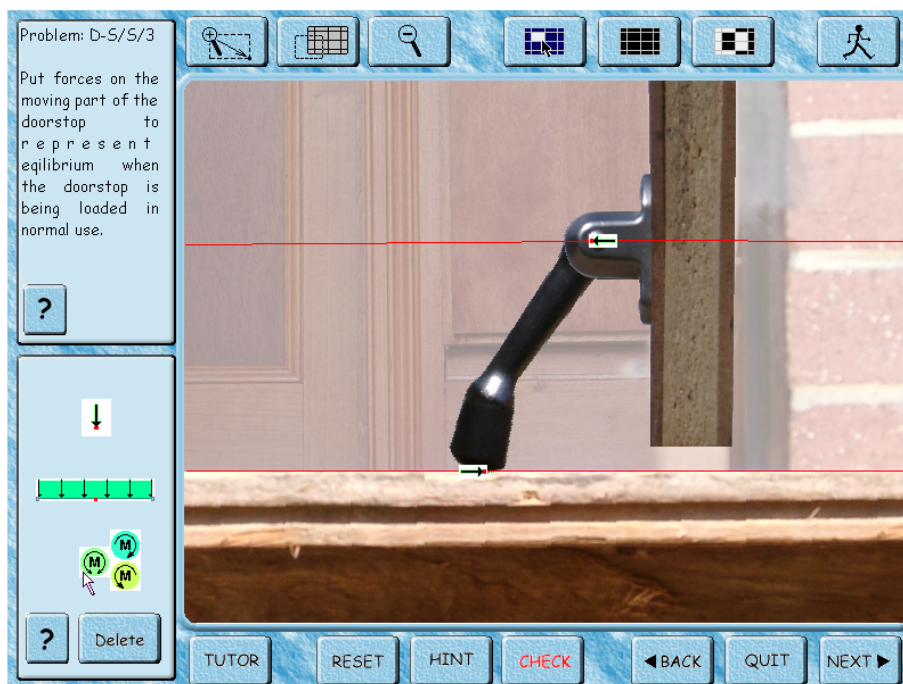


Figure 3: Appearance of MOMUS Tutor screen during an attempt to solve a problem

Through the top row of buttons the student has access to tools that allow the machine to be ‘operated’ (animated or re-configured), and parts to be ‘selected’ (highlighted) or made transparent so that the inner workings can be observed. Other buttons allow the user to zoom in or out so that small assemblages, or the whole machine in its context, can be viewed. Students are able to construct the line-diagram models that represent the machine or selected portion of the machine by dragging and dropping segments of the model onto the appropriate part of the image(s). A typical tutorial problem would ask a student to construct a line diagram model for some part of the image under defined external conditions (e.g., loads, temperatures, speeds) that might be used in the solution within a particular engineering science (e.g., dynamics, thermodynamics, control). Their answer will comprise several components, including the machine configuration at some point in its cycle, the highlighted components, and the locations, shapes and alignments of the various modeling ‘icons’ that define the model. The answer is therefore essentially a unique 2-D image, the construction of which is rendered in a convenient manner by the Tutor’s interface.

Figure 3 shows the screen of the Tutor during the formulation of an answer to a basic problem in statics. The object (in this case a simple doorstop) fills the main window. The task is defined in the upper left-hand window, and the modeling icons (point and distributed forces, and moments) are available in the lower left-hand window. When the Tutor offers feedback after a student asks it to ‘Check’ the answer, a feedback window overlays modeling icons. The top row of buttons allow the image to be manipulated – zoomed, selected and animated, and the lower row of buttons allow the problems to be navigated. The student’s answer (in the case shown in Figure 3) is two copies of the point force icon dragged, dropped and rotated onto the image, which now contains several de-selected parts of the machine, defining the free-body boundary.

The Tutor is programmed to diagnose the student’s answer, and then to offer appropriate comments that have been prepared in advance by the educator who set the problem after switching the software to an authoring mode.

While authoring, the educator has the opportunity to create a number of possible ‘solutions’ – correct or incorrect, that are judged to be likely responses by novice designers. The first solution that the educator creates is defined as the ‘target’ solution (the most desired correct solution), but any successive solutions loaded into the Tutor can be examples of the most common types of errors that students tend to make. For example, the solution shown in Figure 3 was an incorrect solution that was offered by 20% of the students who attempted the problem when it was set on paper as a ‘spot test’ during 2002! This approach follows a similar philosophy to that adopted by Scott and Stone (1998) with their introductory Dynamics tutor at the University of Western Australia and their generalised ‘Jellyfish’ tutorial environment. Using the set of authored ‘solutions’, the diagnosis in the MOMUS Tutor is performed in two stages.

First stage: The diagnosis conducted by the Tutor is a search through its set of stored solutions for a close match (within author-selectable tolerances), and, if it finds a match, the Tutor offers the corresponding feedback comment that was pre-stored along with that solution.

Second stage: If a close match to the student’s answer is not found, the Tutor uses its second diagnosis routine. In this routine the Tutor compares successive elements of the answer with the ‘target solution’, and offers feedback associated with the first substantial mismatch that it finds. These feedback comments were also pre-stored when the educator set the problem, and cover circumstances such as ‘incorrectly selected parts’, ‘inappropriate icons’, ‘missing icons’, and wrongly placed, sized and rotated icons.

The Tutor keeps track of the number of times that an identical ‘error’ occurs, and provides access to second and third level ‘hints’ that the educator has prepared. The student has no direct access to the ‘target’ solution, or any other ‘good’ solutions that have been stored, so the hints and feedback have to be constructed by the educator to direct students toward the target, and the target solution needs to have a feedback comment that identifies itself as the termination of the problem. In this way it was intended that the Tutor could follow a similar structured approach to that of an experienced human personal tutor.

Authoring in MOMUS Tutor

The access point for problems in the Tutor is a ‘contents page’. This page (Figure 2) displays a grid, where the rows represent the alternative ‘machines’ available for analysis. The ‘doorstop’ in Figure 3 is one of these machines. The columns represent the engineering sciences for which problems may be authored. The ‘static equilibrium’ icons in Figure 3 belong to one of the engineering sciences. It is therefore possible to set or access problems in any of the nominated engineering sciences applied to any of the machines, by selecting the corresponding grid element. The Tutor can be used to create, then access up to nine problems in each grid element, although it starts from a completely empty grid. Currently the grid is 8 machines x 6 sciences, allowing access for $8 \times 6 \times 9 = 432$ separate problems.

After entering the authoring mode, protected by a password, the educator can select any of the grid elements to create or edit a problem. This route is shown down the right hand side of Figure 1. The starting configuration is then chosen: image size, scales, default sensitivities (tolerances) for the diagnosis, the problem text and the subset of modeling icons, including any pre-placed icons if desired. The feedback comments associated with the second stage

diagnosis are then entered, followed by the target configuration and its specific comments. The required diagnostic accuracy (tolerance) for this, and other sample solutions can be set for each solution by manipulating visual ‘tolerance zones’. For icons that can be rotated, the tolerance zones for the alignment are shown as sectors of a circle, such as those shown as dark pink associated with a ‘beam’ icon in Figure 4: the Tutor will accept any alignment of the icon that falls within the sector. The tolerance zones for positions of icons in x-y space are rectangular areas, such as those associated with the forces in Figure 4. Other icons may have special characteristics: the length of the beam in Figure 4 can be no less that that shown, but could be larger (with redundant overhang), so its tolerance zone for length is indefinitely long each side of a central minimum length.

Any number of alternative solutions and their feedback comments are then entered. The problem-setting task is then terminated, and all of the information about the problem and its solutions is recorded in a separate text file, averaging 35 kilobytes in size (and easily transmitted through the internet).

Subsequently, when the Tutor is opened, it searches its default directory for problem text files, and, finding any, makes them accessible in the contents page. In the authoring mode, any existing problem can be edited or extended: in the tutoring mode, each problem can be selected individually, or in sets, and attempted by students.

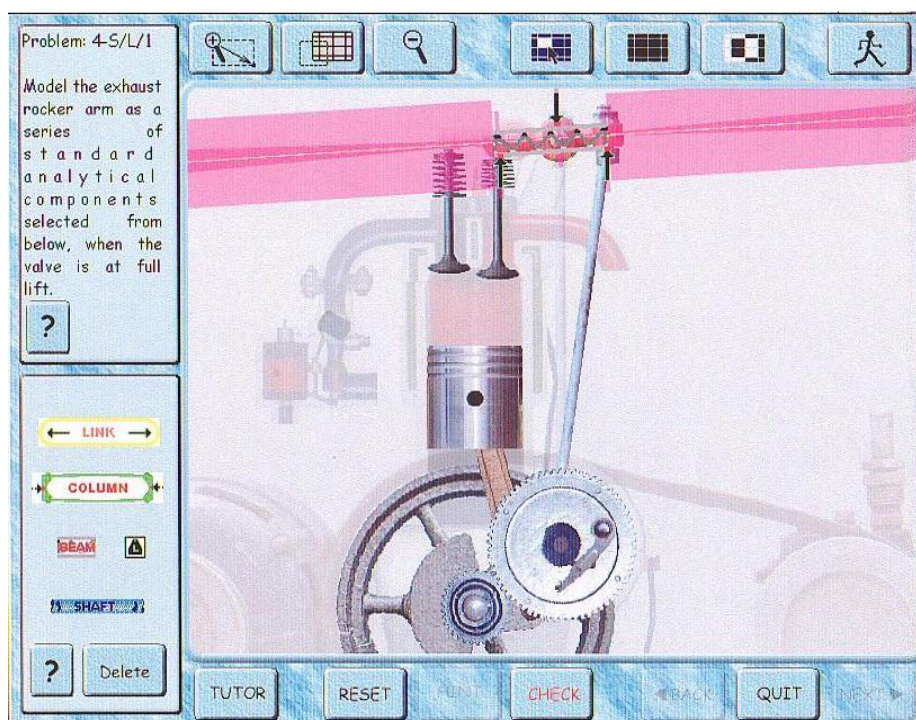


Figure 4: Authoring a problem, showing the pink tolerance zones associated with machine element icons (beam and point forces)

Evaluation of MOMUS Tutor

The development team completed the coding of the core parts of the MOMUS Tutor in 2001. At the end of 2001, only one piece of representative machinery had been included, and only the set of static equilibrium icons. Nevertheless, most of the desired characteristics of the software had been completed. This included the methods of manipulating the images,

manipulating the icons through pop-up selections, rotations, and distortions, diagnosing the answers, and authoring new problems. It was therefore possible to prepare up to nine problems in statics with one piece of machinery, and to test the software with novice designers. The Tutor was mounted onto file servers at university A, and onto a smaller number of stand-alone PC's at university B.

The evaluation comprised the following sequence:

- 1 Conceive a problem that could be set in the MOMUS Tutor, and set the problem on paper.
- 2 Administer the problem to students at the respective universities, allocating credit points for correct solutions. These problems, which only required the placement of two or three force images, were intended to require only about 5 minutes of effort.
- 3 Collect the alternative solutions and group them into identical (or near-identical) sets.
- 4 Code the most common of the sets of solutions into the Tutor, along with the associated feedback comments. (Across the four problems set during the first half of 2002, there was an average of 18 different sets of solutions coded per problem. This coding took an average of 1.5 hours per problem, following an average of 1.5 hours to define each set from the 300+ students' attempts on each set).
- 5 Encourage some students to seek the solutions to the problems via the Tutor.
- 6 Administer a similar problem, and dissimilar problems involving the same principles, to all students, and seek differences in the success rate between students who have used the software, and those who have not.

Results of the evaluation

Four different problems on the equilibrium of the parts of the doorstep machine were administered on paper during the first half of 2002. These included the basic, 2-force single moving part through to the more complex three-force 2-part doorstep assembly. Students at the two universities attempted these problems simultaneously. Because of earlier experiences (Field, Burvill and Weir 2001) the authors were not surprised at the low success rate of their students: only 1%-5% of the students created correct solutions to each of the tasks.

The most common solutions for each problem were coded into the Tutor and students at university A were given access to those solutions one week after they had attempted the problem. For a variety of reasons, only a few students took the opportunity to explore the solutions and find out how well they had performed, or to seek the 'correct' solution.

Following the fourth problem, a fifth test problem in equilibrium was set, representing an abstract 2-piece object with one external load, and two support points. The abstract object could be analysed with exactly the same set of force images as was one of the four tasks set on the doorstep, but the similarity would not be immediately obvious to a novice designer. Students were also asked to indicate how much time, if any, they had spent using the Tutor software during the previous month.

Although only a small number of novice designers at university A indicated that they had used the Tutor, 50% of this group reached the correct solution for the fifth problem, whereas only 5% of the remainder of the novices did so (consistent with the capabilities of the group found in earlier tests). This was not conclusive evidence that the Tutor had increased student

skills in the area, but at least the results were encouraging. An alternative explanation: the self-selection of students who used the Tutor may have biased this group to contain more educationally-motivated students, who may well have found alternative sources of learning. Ethical and administrative obstacles precluded the authors from using fully randomised groups.)

In the main evaluation study, students at university B were not given access to the Tutor until the classroom tests had all been completed. However, their final examination in the design subject was to include a fifth doorstep equilibrium problem, another more abstract problem in static equilibrium (comprising a multi-segmented loaded ring), and a set of questions relating to their use of the Tutor. Four more potential doorstep problems were coded into the Tutor, making a total of eight, and students were told that one of the four new problems would appear on the examination. None of those four new problems contained the correct solution, or useful comments if students attempted to solve them in the Tutor. It was expected that some students would try to use the Tutor on the first four problems before they accessed the four new problems, but that some students would rely on others to ‘find’ the new problems for them, and therefore not access the Tutor at all.

The examination results were analysed to distinguish the achievements of those who had used the Tutor from those who had not. The results indicated a significant correlation of 0.33 between the number of problems solved using MOMUS Tutor and success on the examination problem. Cross correlations with other possible causes for differential performances were not significant. (For example, there was no significant correlation between the success on the examination problem and success on the test problems throughout the semester, nor between success on those test problems and usage of the MOMUS Tutor). It was concluded that the most likely cause of better examination performance was the successful exposure to problem-solving with the Tutor software.

Discussion and comments

The encouraging findings from the evaluations of the Tutor led to minor refinements in the diagnostic routines and the expansion of the hardware and icon sets to include a four-stroke engine and the elements used for representing columns, beams, shafts and tensile members (Figure 3). The engine image can be animated continuously, or stopped in various critical configurations. By de-selecting (ghosting) external components, such as the crankcase, images of all the important separate parts can be seen, selected and magnified for detailed study. These new aspects to the Tutor allow the generation of both static equilibrium and structural elements for both pieces of hardware. During 2003, it is intended that some of these combinations will be tested in a similar manner to those reported earlier.

In separate projects, groups of senior undergraduate designers have identified hardware that would be motivating to junior designers, and have identified the types of modeling tasks that have been found most difficult. These have included the subtleties of dynamic and kinematic analysis, and the selection of manufacturing processes.

The universities are also supporting a refinement of the process for setting up the initial student attempts, by eliminating the need for paper-based ‘tests’. An on-line ‘Agent’ (Juan et al 2002) will capture student test submissions generated from within MOMUS Tutor, and feed them in summarised form to the educator. Student attempts may then be assessed, and accessed through the ‘edit’ feature in MOMUS Tutor, where the feedback comments can be

appended. When students then re-visit the test problem as a tutorial task, they will have the opportunity to work toward the solution. The Agent will again be able to capture student responses to the feedback comments, feeding them back to the educator, who will be able to determine if the feedback appears to be misleading, or perhaps add intermediate solutions to the set of student responses so that more efficient learning takes place. The overall aim of this latest MOMUS Tutor development effort is to reduce the gulf that exists unavoidably between educator and student in computer-mediated learning environments.

The encouraging outcome from the evaluations has also led to a separate project at University B to use the basic shell of the MOMUS Tutor with a special set of photographic images of various mechatronic devices, along with new drag and drop labels, to provide an introduction to the separate discipline of Mechatronic Engineering. There is a long-term plan to extend this approach into other engineering and non-engineering disciplines where convenient customisation, author accessibility, and immediate student feedback on modeling tasks are desirable educational goals.

Conclusion

The electronic tutor gave valuable learning experiences to the students who used it in the solution of classic problems in static equilibrium, and assisted in improving a universally weak skill. The expansion of the Tutor to include a wider range of modeling icons, and more exciting machinery, is under way.

References

- Ferguson, E.S., (1992). *Engineering and the mind's eye.*, Cambridge MA: MIT Press.
- Field, B.W, Burvill, C.R and Weir, J.G., (2001). Student misconceptions in engineering design. *Proceedings of the International Conference on Engineering Design '01 (ICED'01), 1*, (pp 253-260), Glasgow, UK.
- Juan, T., Pearce, A and Sterling, L. (2002). ROADMAP: Extending the Gaia methodology for complex open systems. *Proc First International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS2002)*, (pp.3-10), Bologna.
- Rosenweig, G., (2000). *Using Director 8*. USA, QUE.
- Samuel, A.E. and Weir, J.G., (1991). The acquisition of wisdom in engineering design. *Instructional Science*, 20, 419-442.
- Samuel, A.E. and Weir, J.G., (1999). *Introduction to Engineering Design*. Oxford: Butterworth Heinemann.
- Scott, N. and Stone, B., (1998). A flexible web-based tutorial system for engineering, maths and science subjects. *Global Journal of Engineering Education*, 2, (1), 7-16