INTERNATIONAL CONFERENCE ON ENGINEERING DESIGN
ICED 03 STOCKHOLM, AUGUST 19-21, 2003

AN ELECTRONIC TUTOR FOR STRUCTURAL DISTILLATION

Bruce Field, Colin Burvill and John Weir

Abstract

For a variety of safety and economic reasons, novice designers in Australian universities have experienced diminishing access to examples of the real artifacts that are the subject of their design tasks. Consequently, their understanding of the theories and analytical tools for design analysis have become abstract, with their studies becoming more disassociated from reality, leading to reduced motivation. In an attempt to bridge the gap between abstract theories and their practical embodiments, the authors have created a software tutor that contains both photographic images of real machinery, and sets of modeling 'tools'. Consequently, academics responsible for teaching the engineering sciences can set tutorial tasks which develop the novice designers' capability of relating real machinery to the abstract models. The individual personality or style of the educator is captured in the authored feedback comments, and a two-tiered diagnostic routine selects the most helpful feedback as the designer works toward a satisfactory model. An evaluation of the software demonstrated that undergraduates who had solved problems with the electronic tutor's aid performed significantly better in formal examinations with similar modeling tasks.

Keywords: Design modeling, structural distillation, CAL, design tutoring

1. Introduction

Engineering designers deal with real artifacts. At various stages of the design process, some artifacts may exist as actual objects, while others will exist only as abstraction of various kinds: perhaps as outline sketches, or block diagrams with symbols, or even just equations. Designers must move their focus between real artifacts and their abstractions[1,2] as they converge onto a solution that will satisfy the constraints in the design brief.

The designer uses the engineering sciences, with their abstract formulations, to model the realistic artifacts that will be the final outcome of the problem. We call the process of transforming the concept into its analysable models that of 'structural distillation'[3].

A designer needs to learn how to perform this distillation when they learn about the models and how to manipulate them. Unfortunately, in many undergraduate engineering courses, this is not achieved. The teacher typically begins with a simplified line diagram (already a model of the artifact), and develops the novice designers' ability to formulate and solve equations that connect the unique set of parameters placed on the diagram.

The authors have observed that the majority of the engineering sciences taught at their universities are bereft of the artifacts associated with the science, with the result that design efforts based on those sciences employ awkward, often incorrect models, and lead to unworkable design outcomes. To address this shortcoming, the authors have created a software tool that is intended to encourage novice engineering designers to learn how to formulate the correct (or adequate) models from images of real artifacts (normally, but not exclusively, photographic images), across a range of the engineering sciences.

This paper describes the underlying educational philosophies of that software, and reports research outcomes from its first year of use. The software is called MOMUS Tutor (MOnash-Melbourne Universities' Structural Tutor), reflecting the collaboration associated with its development. (Momus is an English word for a fault-finder or persistent critic, derived from the Greek god of ridicule).

2. Principles of MOMUS Tutor

In the version of MOMUS Tutor (the Tutor) coded in 2001, each real machine was represented by a set of photographic images, constructed, where appropriate, as an orthogonal view. Machines are made from parts, and the parts are separate images, 'assembled' to show the whole machine. The learner has access to tools that allow the machine to be 'operated' (animated or re-configured), and parts to be 'selected' (highlighted). Other tools allow the user to zoom in or out so that small assemblages, or the whole machine in its context, can be viewed.

Novice designers 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). A novice designer's answer will comprise several components, including the machine configuration at some point in time, the highlighted components, and the locations, shapes and alignments of the various modeling 'icons' that define the model. Their answer is therefore essentially a unique 2-D image, the construction of which is rendered convenient by the Tutor's interface.

Figure 1 is 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 1) 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.

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 had the opportunity to create a number possible 'solutions' – correct or incorrect, that were judged to be likely responses by novice designers. The first solution that the educator created is defined as the 'target' solution (the most desired correct solution), but any successive solutions loaded into the Tutor could be examples of the most common types of errors that students tend to make. For example, the solution shown in Figure 1 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'! . Using the set of authored 'solutions', the diagnosis is performed in two stages.

The *first stage* of 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.

However, if a close match to the student's answer is not found, the Tutor uses it's *second stage* 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.

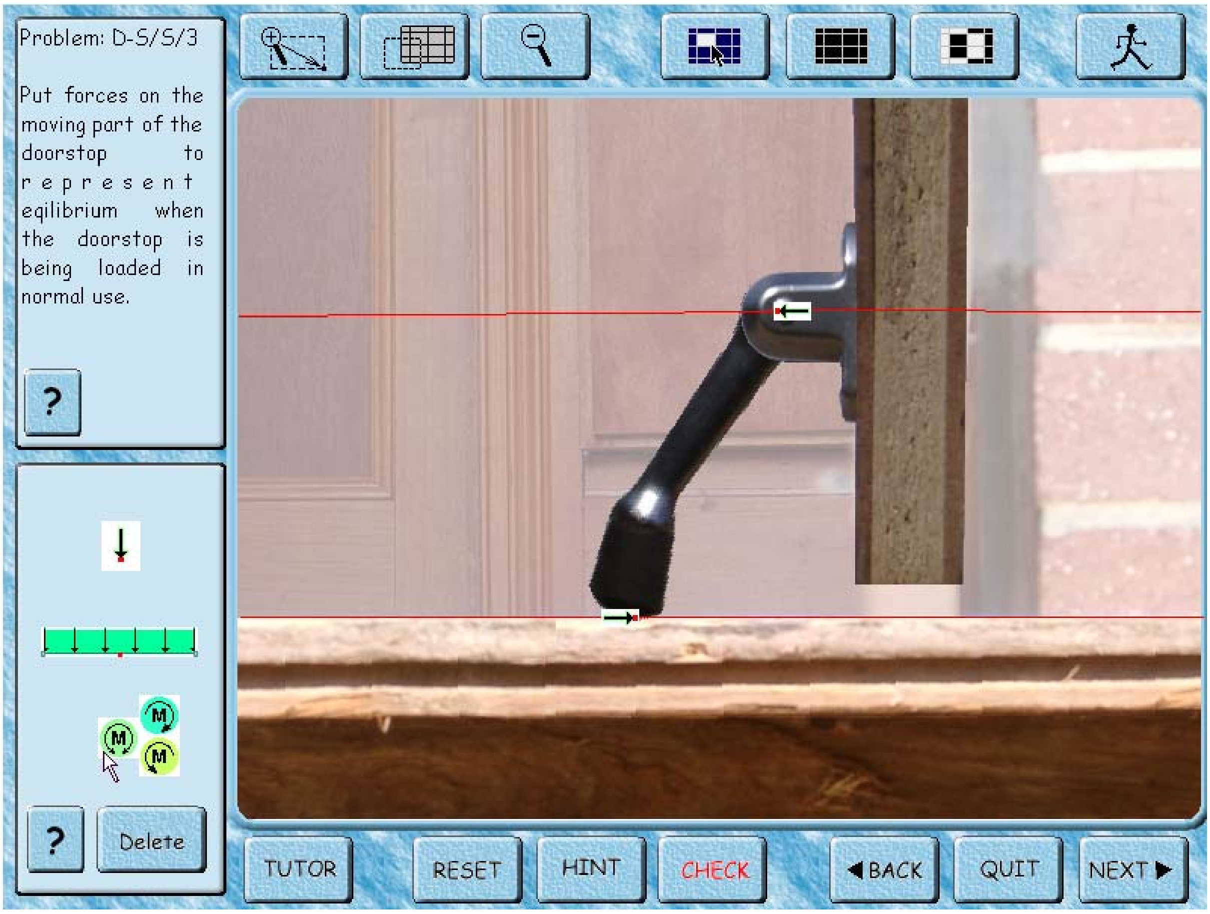

Figure 1. Appearance of MOMUS screen during an attempt to solve a problem

3. 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 1 is one of these machines. The columns represent the engineering sciences for which problems may be authored. The 'static equilibrium' icons in Figure 1 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 8x6x9 = 432 separate problems.

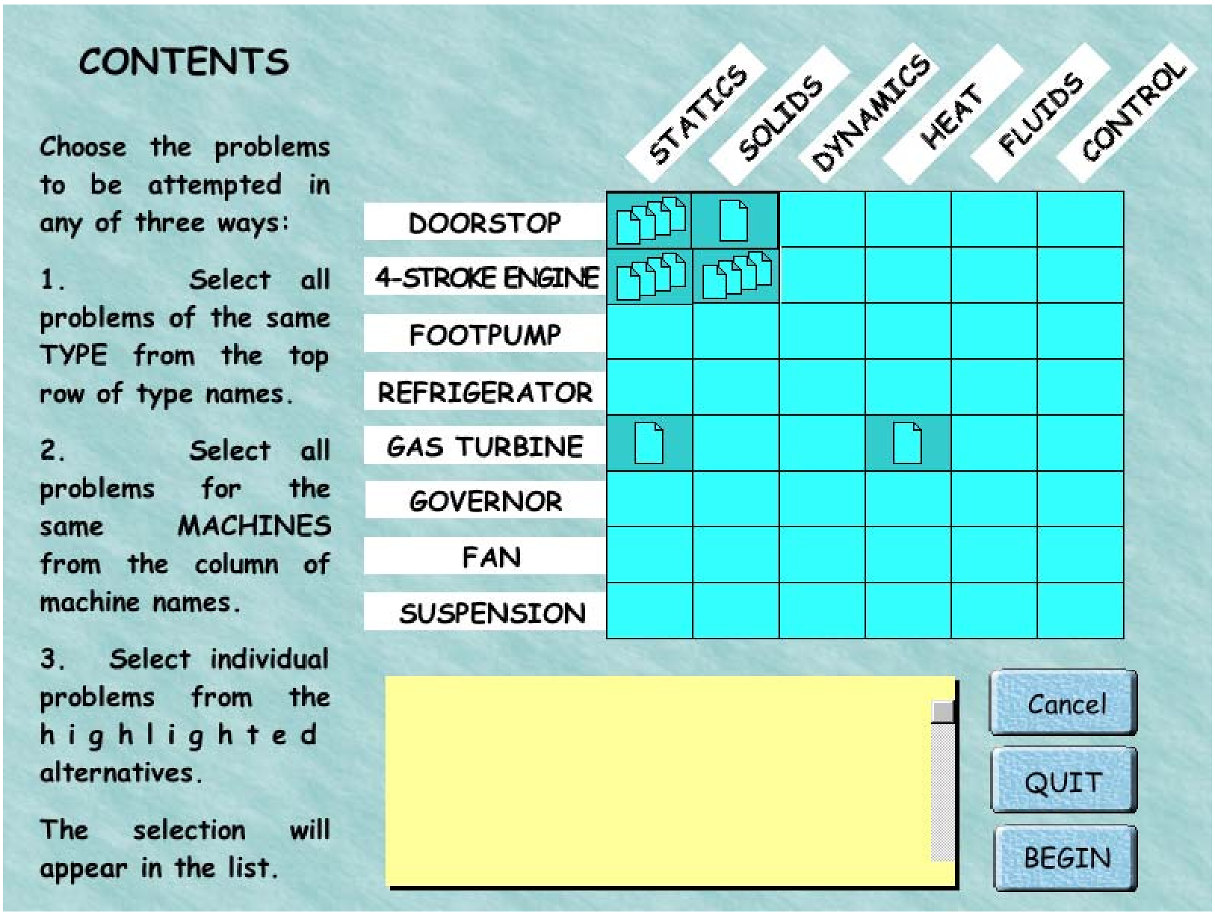

Figure 2. Contents page of MOMUS Tutor with several problems available

After entering the authoring mode, protected by a password, the educator can select any of the grid elements to create or edit a problem. 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 3: 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 3. Other icons may have special characteristics: the length of the beam in Figure 3 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.

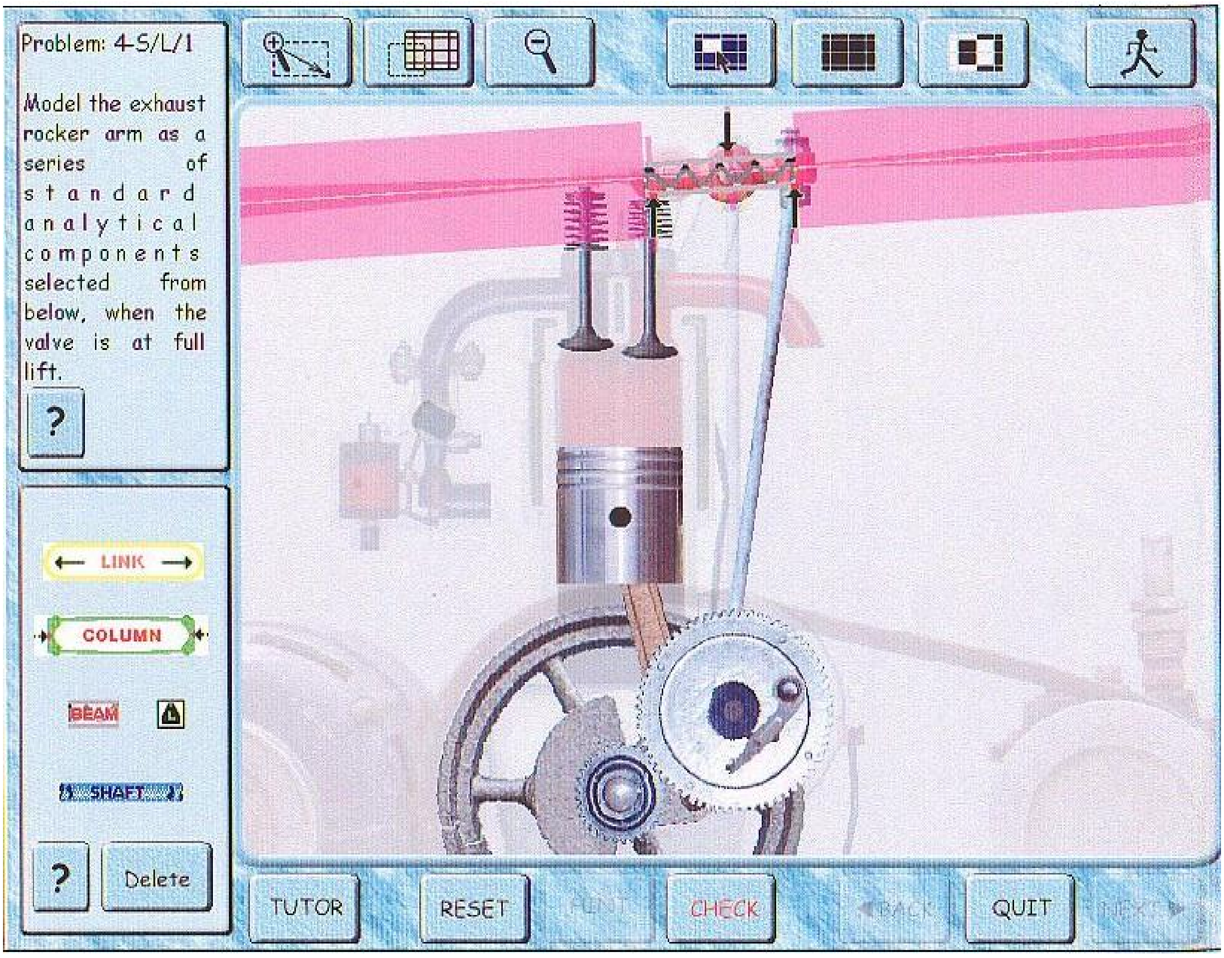

Figure 3. Authoring a problem, showing the pink tolerance zones associated with machine element icons

4. 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.

Familiarisation modules had also been written for MOMUS Tutor, allowing students to access the Tutor without separate instruction. The Tutor then comprised three main ‘movies’ written with Macromedia’s Director 8 (commonly used for coding educational software). One movie opens the software and gives access to the other two: the first being the familiarisation routine, and the second being the problem-solving movie. Several Director ‘casts’ are associated with one or more of these movies. Problem text files were authored as needed. The large number of photographic images, and the level of resolution required for realistic imaging, meant that the Tutor with its support files was too large to be available over low-speed telephone-line modems. Consequently, the Tutor was mounted onto fileservers at university A, and onto a limited 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.

5. Results of the evaluation

Four different problems on the equilibrium of the parts of the doorstop 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 doorstop assembly. Students at the two universities attempted these problems simultaneously. Because of earlier experiences[4], 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 doorstop, 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 is that 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 us 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 doorstop 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 doorstop 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.

6. 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.

In a separate project, the universities are 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’[5] will capture student test submissions generated from within MOMUS Tutor, and feed them in summarised form to be accessed by the educator. Then the attempts may 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 environment.

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.

7. 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

[1] Ferguson, E.S., “Engineering and the mind’s eye”, MIT Press, Cambridge MA, 1992.

[2] Samuel, A.E. and Weir, J.G., “The acquisition of wisdom in engineering design”, Instructional Science, Vol. 20, 1991, pp.419-442.

[3] Samuel, A.E. and Weir, J.G., “Introduction to Engineering Design”, Butterworth Hinemann, Oxford, 1999.

[4] Field, B.W, Burvill, C.R and Weir, J.G., “Student misconceptions in engineering design”. International Conference on Engineering Design ‘01 (ICED’01), Vol. 1, Glasgow, 2001, pp 253-260.

[5] Juan, T., Pearce, A and Sterling, L. “ROADMAP: Extending the Gaia methodology for complex open systems”, Proc First International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS2002), Bologna, 2002, pp.3-10.

Corresponding Author

Bruce Field Department of Mechanical Engineering, Monash University, Victoria 3800 Australia
Tel: Int +61 3 9905 3518 Fax: Int +61 3 9905 1825 E-mail: bruce.field@eng.monash.edu.au