Global J. of Engng. Educ., Vol. 2, No.3
Printed in Australia

The *ASTutE* Tutorial Assistant: Efficient, Accessible and Interactive*

Fiona Austin
Steve Rothberg
Department of Mechanical Engineering, Loughborough University, Loughborough, Leicestershire, LE11 3TU, United Kingdom

The *ASTutE* project provides a computer-based tutorial resource designed to work alongside tutors to help meet the challenge of teaching increasing numbers of students with increasingly diverse backgrounds. *ASTutE* is used to solve the majority of student difficulties, freeing the academic tutors to deal with more complex misunderstandings to which they are currently unable to devote sufficient time. Problems are broken down into key stages and then substages using question types, including multiple choice/response, text, number-unit and maths entry, diagram creation, hot spot and drag-and-drop. A key feature of *ASTutE* is the ease with which tutors can quickly write new, or modify existing, problems by editing a simple template. Templates are edited in a way that demands only the most basic knowledge of any spreadsheet package. Students use *ASTutE* interactively to check answers or as a detailed help system, accessing only as many help stages as they require. A fundamental objective of *ASTutE* is to provide feedback tailored according to the responses of an individual student. As a result, the student takes one of many possible routes through a tutorial problem, consistent with their understanding of the topic.

INTRODUCTION

In recent years, UK student numbers have grown substantially, while, at the same time, the backgrounds of these students have become increasingly diverse. Academics are finding that traditional teaching methods alone are struggling to cope with these increased demands and that they must develop new strategies. One option is to incorporate new learning technologies such as Computer-Aided Learning (CAL) into the curriculum to support and enhance existing teaching methods. Engineering, which has a substantial mathematical content, is particularly suited to a computer-based teaching approach and this has led to the development of a significant number of CAL packages in this subject area.

Tutorials are particularly affected by the pressures described, yet experiential learning through tutorial work is an essential component in the consolidation of students' understanding of lecture material. Traditionally, sheets of problems are handed out and tutorials on these problems are held in small groups with students discussing and solving their difficulties with the help of a tutor. But the *small* group is increasingly not so small and *discussing and solving difficulties* on an individual basis is becoming less feasible. The aim, therefore, is to utilise CAL material as an additional tutorial resource in a way that concentrates on making tutorials more effective for all involved.

To this end, several key criteria were identified:

- The tutorial resource must be an integrated part of the curriculum, available at all times for use both in and outside time-tabled sessions.
- It should solve the majority of difficulties for the majority of students, freeing tutors to deal with more complex misunderstandings. Solving every conceivable difficulty would not only be too ambitious, but also inefficient.
- The student interface must mirror the pencil-paper approach as far as possible: defining the problem

* An expanded version of a paper presented at the *1st UICEE Annual Conference on Engineering Education*
* This paper was awarded the UICEE silver award (fourth grade), by popular vote of Conference participants, for the most significant contribution to the field of engineering education

requirements, setting up diagrams, writing and solving equations.

- Students must find it easy to use and be able to work at their own pace, taking the learning route that suits them best.
- It should be possible to incorporate the resource into existing courses without major modification, retaining the use of existing problem sheets.
- Tutors should be able to write or modify problems, which must be a quick and easy process, with no requirement to install special software and a minimal learning curve.

BACKGROUND

There are many excellent existing CAL products in the mathematics and engineering disciplines. These can be either commercial or non-commercial and include diagnostic testing products, teaching and learning materials and assessment environments. The most relevant of these were considered for use in *ASTutE* (*A*utomated *S*tudent *Tut*orial *E*nvironment).

A non-subject specific product considered was Question Mark, commercial software that allows creation, delivery and marking of questions on a computer in a very simple and easy manner [1]. Available question types are multiple choice/response, hot spot, selection, number/text entry. Questions are mainly presented sequentially with textual feedback available, but simple jumps between questions can also be added. Universities have used Question Mark for diagnostic mathematics testing, but other products have been developed specifically for this purpose [2][3]. One example is DIAGNOSYS, a DOS-based system funded by the Teaching and Learning Technology Programme [4-6]. Single stage, multiple choice, numerical and simple algebraic questions can be written.

Much CAL software consists of teaching material followed by sets of questions, and Mathwise and CALMAT fall into this category [7][8]. CALMAT has a sister product, TASMAT, a tutorial and assessment system that enables tutors to construct customised tests with random parameters [9]. Records of student results are maintained and these may be inspected and printed out by the tutor. The logging on, reporting and monitoring facilities of this software package are suitable for use in *ASTutE*. Other packages that consist only of questions are AIM, from the CALM project [10-12], and Maths Assessor, part of the Intermath project [13][14]. Maths Assessor delivers sets of questions and collects student responses. Questions are prepared in a conventional Windows word-processor and can contain diagrams and mathematical notation. The maths entry tool developed during this project has been adopted by *ASTutE*.

It was always the intention to incorporate existing code where possible. This was also the case for the unit-number checking code developed by Dynamic Teaching Solutions, a project that provides software and hardware resources for teachers of engineering dynamics, including a web-based tutorial system [15][16].

In summary, existing CAL materials tend to fall into one of three broad categories:

- Pre-written questions with no facility for tutors to write their own questions.
- A package allowing tutors to write problems easily and quickly but limited to single stage, mainly multiple choice or text match question types.
- A package allowing tutors to write more complex problems, but still mostly single stage and often requiring special editing software with an associated learning curve.

A key objective of *ASTutE* is to allow tutors to write multiple stage, multiple route problems easily and quickly, without the need to learn special software. To avoid unnecessary duplication of effort, it was hoped that an existing product would meet, or could be modified to meet, the desired criteria. While this was not the case, the knowledge gained through consideration of the existing material greatly influenced the development of the *ASTutE* Tutorial Assistant and elements of existing products have been incorporated where appropriate.

THE *ASTUTE* TUTORIAL ASSISTANT

ASTutE is flexible and designed to be used in different ways according to students' needs and abilities, as described below and shown in Figure 1.

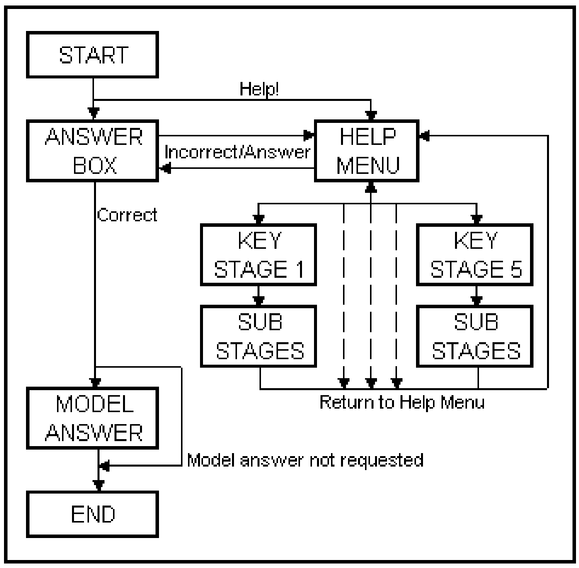

Figure 1: Flow chart of basic *ASTutE* structure.

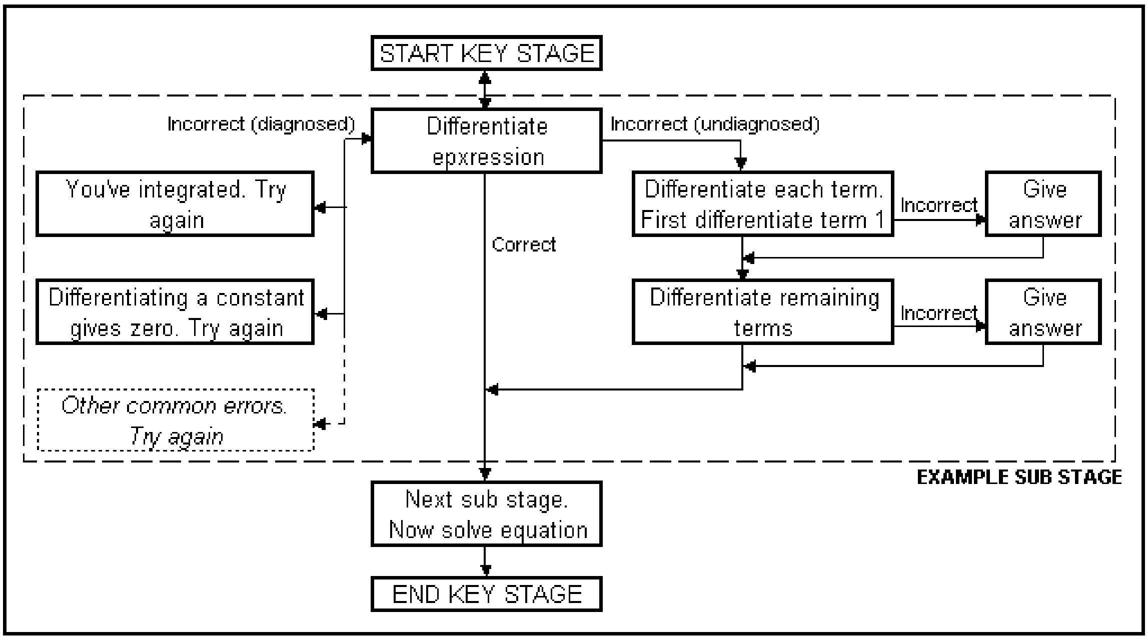

Figure 2: Example substage.

The initial screen of *ASTutE* presents students with an answer box in which to enter a final answer if access to the help system is not required (this is demonstrated in Case Study 1). Therefore, in the simplest scenario, *ASTutE* is used as an answer checking system. Strong students who have successfully completed a problem on paper may log on simply to check answers and perhaps look through the model solution.

If the help system is accessed, students choose any key stage of help they require. The second scenario is when students experience a difficulty at a certain point in a problem or are uncertain about their chosen method, for example, choosing equations. Here, *ASTutE* provides specific help, solving the student's particular difficulty without going through the whole problem. In the third scenario the help system is accessed from start to finish. Weak students are led interactively through a problem, stage-by-stage and in a logical manner.

It would be too ambitious and inefficient to attempt to solve every conceivable difficulty for every student and so the tutor is available to address the more complex or unusual misunderstandings.

Problem structure

ASTutE is designed to allow students to tackle a problem in much the same way as they would on paper. This generally consists of setting up the problem data, creating the diagram if applicable, choosing equations to use, applying those equations and solving the equations. Each problem is broken down into five key stages (shown in Case Study 1), each of which can have any number of substages.

A student can work through as many of the five key stages as are felt necessary and the key stages can be accessed in any order. The response to any substage within a key stage determines the substage shown next. Feedback is tailored to the individual responses, resulting in a student taking one of many possible routes through a problem, depending on their understanding of a topic. Students typically make one of a limited number of possible mistakes at any stage in a complex problem and so the necessary substages can be identified relatively easily by a tutor. If an unpredicted but common difficulty emerges, the tutor modifies the existing template to incorporate it.

Figure 2 shows an example of the first substage within a key stage asking students to differentiate an expression. There are many possible routes through this substage. For example, a student who differentiates the expression correctly, moves directly onto the next substage, which is to solve the resulting quadratic equation, taking the most direct route through the key stage. A second student, however, integrates instead of differentiates and so receives appropriate feedback. On a second attempt, the answer is still incorrect but this time the error cannot be diagnosed. The differentiation is now tackled term by term and when finally achieved, the student moves onto the next substage.

Both students finally arrive at the same point but have taken very different routes through the problem. This exemplifies how *ASTutE* consists of multiple stage, multiple route problems. The tailoring of feedback and the multiple route approach encourage active engagement and, therefore, a deeper approach to learning.

Question types and diagnosis

ASTutE supports an extensive range of basic question types, including multiple choice/response, text/number entry, hot spot and drag-and-drop. Below are some of the complex question types that are created from these basic question types:

- *Diagram creation*: drag-and-drop is implemented, allowing flow charts to be built up and diagrams labelled. An example of this question type being used in chemical engineering is given in Case Study 2.
- *Number-unit entry*: students must provide a suitable unit for a numerical answer. This code was developed by the Dynamic Teaching Solutions project and is currently utilised in both mechanical and chemical engineering [15][16].
- *Maths entry*: the simple number entry question type has been complemented with the facility to enter complex mathematical expressions in an easy and familiar way, eg the answer to a differentiation [13]. The maths entry dialogue box used is shown in Figure 3.
- *Multiple response*: although this is a very straightforward question type, the diagnosis is complex. *ASTutE* allows the tutor to give specific feedback to any multiple response combination offered by the student.

Writing problems

The *ASTutE* engine runs as an executable file, reading in problem data from text files. Only these text files need to be modified to write problems and this is done using any spreadsheet package (Excel is used in Case Study 2). The tutor opens up a problem template and then follows the instructions in the file to write a problem. The completed file is then saved in text file format. Tutors need not learn to use any special software to be able to write problems. This is very important in encouraging the uptake of this new teaching tool.

The only requirement for someone wishing to write their own problems is a basic spreadsheet knowledge. Both the *ASTutE* engine and associated problem templates were developed in close collaboration with tutors to ensure applicability and ease of use.

ASTUTE CASE STUDIES

The first case study illustrates the principal generic features of the student interface: the initial screen and help menu. The second case study gives an example of a complex question type, including how this is created by a tutor.

Case study 1: partial fractions

The Maths Learning Support Centre in the Department of Mathematics provides a range of services designed to support any student in the University in their learning of mathematics. A range of facilities are available to achieve this aim, including handouts, textbooks, videos, computer-based materials, personal guidance and one-to-one tuition. *ASTutE* is one of the computer-based materials utilised.

One common problem area in mathematics is the understanding of partial fractions and this can be tackled through *ASTutE*. Students are encouraged to attempt problems on paper and then to use *ASTutE* for

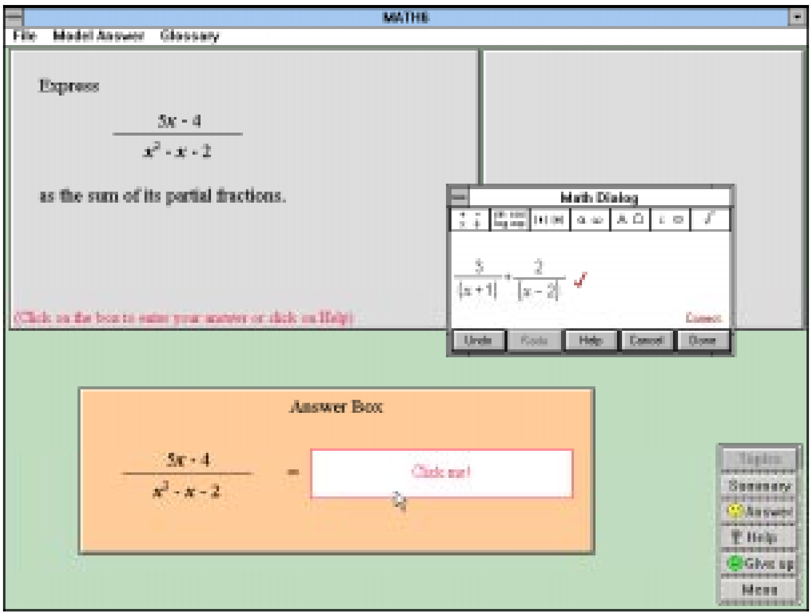

Figure 3: Partial fractions: initial *ASTutE* screen.

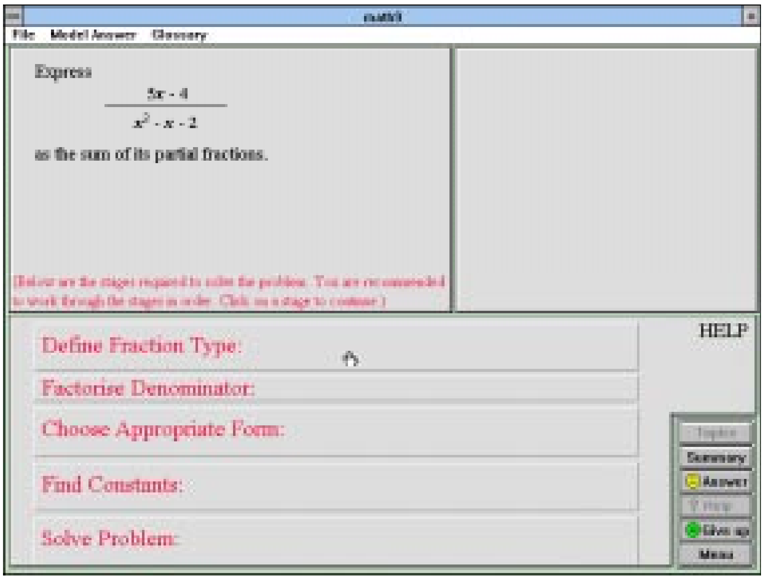

Figure 4: Partial fractions: help menu.

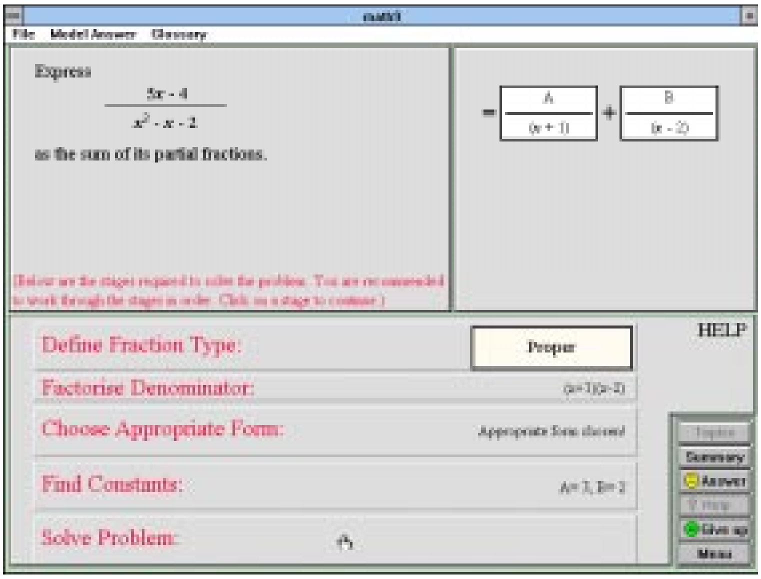

Figure 5: Partial fractions: help menu with four out of five of the summary stages completed.

either answer checking or as a detailed help system.

After *ASTutE* has been launched and a partial fractions problem selected, the student is presented with the initial screen shown in Figure 3, which displays an answer box encouraging the student to enter the final answer. The answer to this problem is a complex mathematical expression and so clicking on the answer box brings up a maths entry dialogue box that allows the student to enter the answer in an easy and familiar way.

If the answer given is correct, as in this example, then the student can move directly onto the next problem, checking a model answer first if desired. This is the quickest route through *ASTutE* and allows it to be used as an answer checking system.

If, however, the student is unable to complete the problem, then they can enter the help system by clicking on the *Help* button in the bottom right hand corner of the screen. The main help menu is then displayed, as shown in Figure 4. The five key stages of this problem can be seen: Define Fraction Type, Factorise Denominator, Choose Appropriate Form, Find Constants, Solve Problem. Each of these key stages may have one or more substages.

Such a structure is also important in encouraging the student to develop a logical approach to problem solving. When a student experiences a specific difficulty in a problem or is uncertain about the chosen method, eg choosing equations, *ASTutE* is able to provide specific help, solving the student's particular difficulty. Students may choose any key stage of help they require and they can, therefore, obtain specific help on a given problem without going through the whole problem.

The weakest students, those who are in need of detailed help, will access the help system from start to finish and will be led interactively through the problem, stage-by-stage. This allows the students to build up a knowledge of how to tackle a problem in a logical manner and will help them with future problems. This can be seen in Figure 5 where a student working through the help system has completed four out of the five stages, as indicated by the summary information shown on the stage buttons. By comparison, Figure 4 shows a screen shot in which none of the five key stages have yet been accessed.

A student's progress through the problem is intended to mirror the pencil-paper approach as far as possible, with many different question types being utilised to achieve this. For example, in the first substage of each key stage, multiple choice was used to define the fraction type, the maths entry tool was used to factorise the denominator, drag-and-drop created the appropriate form, and simple number entry specified the constants. Tailored feedback is offered to incorrect responses at each help stage of the problem, with each student taking a different route to gain a full understanding of the problem.

Case study 2: recycles

The Department of Chemical Engineering uses *ASTutE* to provide additional tutorial assistance for their Material Balance module. Tutorial problem sheets are handed out in the traditional way, but students attend tutorials in a computer lab where they have access to *ASTutE* to solve the majority of their problems, while a tutor circulates the room to solve the more complex problems. A particular difficulty encountered by students across all engineering and science disciplines is to create the relevant diagram from information given in the problem text. *ASTutE* allows the student to create the problem diagram using individual components from a library and gives detailed feedback throughout the process. This builds up students' confidence in conceptually difficult areas as they develop their own models of understanding.

The first crucial step in the Material Balance problems is to create a flowsheet from the given problem information. This flowsheet is then used as the basis for solving the rest of the problem. The screen-shot of Figure 6 shows a student setting up the flowsheet for a chemical process that consists of reacting a given substance, removing the product and recycling any unused substance. The student chooses a component from the library in the bottom half of the screen and places it anywhere on the grid in the top right hand corner. Another component is then selected and gradually the flowsheet is created. Components can be repositioned or removed at any stage and the student receives instant feedback as components are placed, prompting, for example, if flow is discontinuous.

When the student is satisfied with the flowsheet drawn, the *Check* button is pushed and more detailed feedback is then given if the answer is incorrect. The tutor is able to customise these feedback messages easily to give specific feedback for specific errors. As students can place components anywhere on the grid, there are usually many possible correct configurations and *ASTutE* is flexible enough to allow this. However, the tutor does not need to have to think of all of these!

Tutors create or modify problems through the *ASTutE* template file, which can be edited in any spreadsheet package. An excerpt from a tutor's problem template file is shown in Figure 7. Tutors are prompted by questions and comments within the template to supply the required information. In this case, there are only two compulsory boxes to complete: *Does the answer contain a recycle?* (yes) and *What is the correct answer?* The second question simply requires the tutor to specify the correct order of the main components; all other ancillary components are dealt with by the *ASTutE* engine and this process is totally transparent to the tutor. There are also other optional boxes where a tutor can specify common wrong errors and supply appropriate feedback. A similar approach applies to other subjects and has been used, for example, to create free-body diagrams in engineering mechanics.

Once the problem has been created or modified, the tutor saves the file and then launches *ASTutE* to check that it reflects their intentions.

CONCLUSION

ASTutE provides tutors with an additional teaching resource that supports and enhances traditional teaching methods by making tutorial assistance more effective. Tutors can quickly and easily write problems and make them available for student use. Students have access to an additional source of help, 24 hours a day. *ASTutE* is interactive and helps students to gain confidence in their work. Feedback is tailored to an individual student's difficulties and students take one of many routes through a tutorial problem according to their ability, understanding and learning style. The *ASTutE* Tutorial Assistant is an efficient, accessible and interactive tool of benefit to both tutors and students.

SPECIFICATION

ASTutE Tutorial Assistant runs on Windows 3.1 and

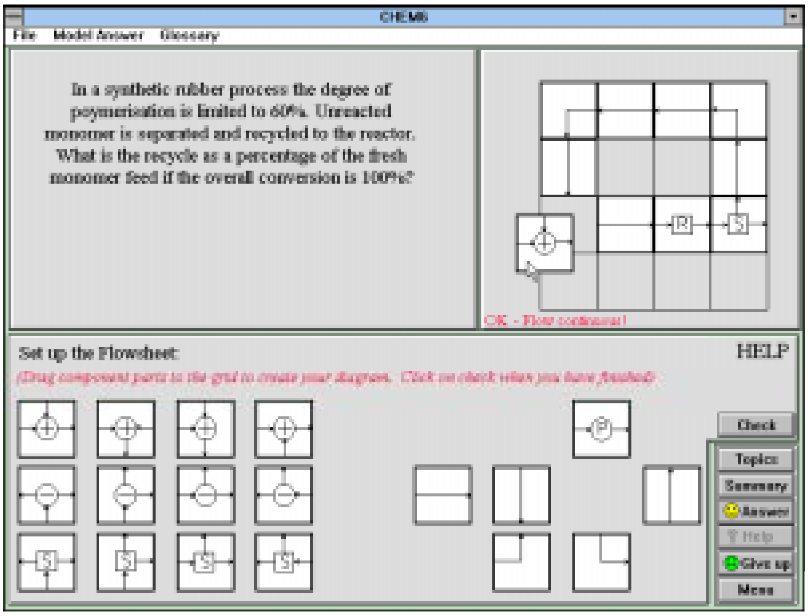

Figure 6: Recycles: creating a diagram.

A. Complete the Process Diagram		
Is there a recycle?	**1**	Set 0/1 for No/Yes.
Correct Answer:	**MixReactSep**	Answer should begin with the Mixer and follow flow direction. Leave no spaces between words.
Wrong answer1:	**MixReactSplit**	Trap up to 3 common wrong answers and give
Feedback1:	**Splitter should be a separator!**	appropriate feedback - leave blank if unused.
Wrong answer2:	**MixPumpSplit**	
Feedback2:	**You shouldn't have a pump in this question!**	
Wrong answer3:		
Feedback3:		

Figure 7: Recycles: example problem template.

95 networked or stand-alone machines. The *ASTutE* engine is written in the multimedia authoring language *Authorware*.

REFERENCES

1. Question Mark: http://www.qmark.com/
2. Greenhow, M. and McCabe, M., Designer software for mathematics assessment. *Maths&Stats*, 6, **1**, 11-16 (1995).
3. Greenhow, M. and McCabe, M., A question mark over mathematics assessment? *Maths&Stats*, 7, **2**, 19-22 (1996).
4. Appleby, J.C., A diagnostic testing facility for basic mathematics. TLTP Newsletter, **3** (1995).
5. DIAGNOSYS: http://www.ncl.ac.uk/~ntltp/
6. TLTP: http://www.icbl.hw.ac.uk/tltp/
7. Harding, R. and Quinney, D., Mathwise and the UKMCC. *Active Learning*, **4**, 53-57 (1996).
8. CALMAT: http://sword.gcal.ac.uk/calmat/
9. TASMAT: http://sword.gcal.ac.uk/calmat/calmat3.html
10. Bull, J., *Using technology to assess student learning*. 20-21 (1993).
11. *AIM: Personal communication with Professor Cliff Beevers*. Heriot-Watt University, UK.
12. CALM: http://www.ma.hw.ac.uk/innovations.html

13. Beilby, M., Making a feature of assessment. *Maths&Stats*, 5, **2**, 6-10 (1994).
14. Intermath: http://www.bham.ac.uk/CBLProjects/im.htm
15. Stone, B.J. and Scott, N., Innovation in teaching engineering dynamics: UWA-CPCS. Copy at: http://www.mech.uwa.edu.au/Dynamics/Promotion.html
16. Dynamic Teaching Solutions Home Page: http://www.mech.uwa.edu.au/Dynamics/default.html

BIOGRAPHIES

Fiona Austin manages the Faculty of Engineering Teaching and Learning Support Centre at Loughborough University in the UK. She joined the Department in 1996, initially as a project assistant in the Department of Mechanical Engineering, after working for Gutteridge, Haskins & Davey Pty Ltd, a firm of consulting engineers in Sydney, Australia. Prior to that, she studied engineering at Oxford University in the UK, sponsored by Mars Electronics International. Current research interests include developing CAL material to make tutorials more effective, and all aspects of implementing CAL, DL and computer-aided assessment into engineering.

Dr Steve Rothberg is a Senior Lecturer in the Department of Mechanical Engineering at Loughborough University, specialising in the areas of Dynamics and Noise and Vibration. After studying as an undergraduate and as a postgraduate at Southampton University, he joined the Department in 1990. He is a Chartered Engineer and currently holds a Royal Academy of Engineering / ESSO Teaching Fellowship. Current research interests in all aspects of vibration measurement, particularly those using laser technology, complement his activity in the area of computer-assisted learning.