Report: CAA in Mathematics Workshop

Over 40 people from 30 institutions attended this workshop for academics who would like computer support in carrying out assessment. Five contributions from presenters who are actively using computer-aided assessment in undergraduate mathematics courses addressed questions such as:

- Can test editors be designed to construct questions suitable for diagnostic, continuous and grading assessment?
- Which question types best test which mathematical skills?
- How can mathematical assessment packages contribute to deep understanding?

These questions will continue to be addressed through discussion on the email discussion list maths-caa. To join this, send the following one-line message to mailbase@mailbase.ac.uk

```
join maths-caa <firstname> <surname>
```

1. WHAM! Web Hosted Assessment of Mathematics with (Extra Sensory) Perception: Michael McCabe, Portsmouth University

Question Mark Perception is a suite of programs for authoring delivering and analysing the results of on-line tests. It has been used in developing maths tests, some requiring Maple and Matlab skills. Part of the Mathletics question bank developed at Brunel University has also been converted for Web delivery using Perception. More recently, a specialist Java question type has been added at Portsmouth to allow random numeric parameters in questions.

Examples of Perception in action can be found via the web version of this presentation.* In addition, the work done at Portsmouth is presented annually at the Loughborough Computer-Assisted Assessment conferences has been published on the Web. See *Proceedings of the 3rd Annual CAA Conference.**

2. ALICE Interactive Mathematics: Theodore Kolokolnikov, University of Ghent, Belgium

Alice Interactive Mathematics (AIM) is a web-based system designed to administer graded tests with mathematical content. Its main features are: use of Maple as the engine and implementation language; several methods of giving partial credit; various feedback mechanisms; randomisation of quizzes and questions; versatility in question and quiz design; extensive grade reporting and monitoring capabilities; ability to collect surveys; web interface for both teacher and student. AIM can be used to administer graded tests, homework or ungraded self-assessment exercises. A case study using vector calculus has been conducted and more courses are being planned for September 2000, including linear algebra, ODEs and precalculus. Preparation is also under way to use AIM to mark a part of the final exam for linear algebra next year.

The slides for this presentation are available on the Web*, and the project site can be viewed at any time.* There is an article describing recent experience at Coventry in authoring AIM questions on page 147 of this newsletter.

3. Assessment over the Web using CUE: Cliff Beevers, Heriot Watt University

Computer-aided assessment has been used on mathematics courses at Heriot-Watt University for many years, and it is a pioneering site for the setting of formal examinations by computer featuring mathematics entry and partial credits within questions. Full details of this presentation appear on page 17 of this newsletter.

4. Auto-interactive Teaching and Assessment System: Boris Mouzykantskii, University of Warwick

ATAS - the Auto-interactive Teaching and Assessment System - has been developed and used with a class of 70 second year Physics students for three years. Problems are typeset in LaTeX with a few extra commands. The electronic version of assessment is implemented as a (set of) Java Servlets. Full details of the system are available at on the Web* in pdf format and as a live example.

5. TAL: Sharing an Open Testing System: Eddie Wilson, University of Bristol

Bristol has set up a large database of multiple choice questions which are graded, timed and can be used to create Web-delivered testing. Full details of this presentation appear on page 20 of this newsletter.

Pam Bishop (p.bishop@bham.ac.uk)
Maths, Stats & OR Network

* links to all these Web sites are included in the Web version of this report at http://www.bham.ac.uk/msor/workshops/maths-caa

Appendix listing further implementations

6. Question Mark have just released the following information about random parameters:

Professor David Hewitt of Monash University, Australia has developed a new method for including JavaScript in QML to generate random numbers for use in questions. The random numbers can be used in mathematical calculations to create questions with answers that are conditional on the calculations of the random numbers in the question stem. To find out how to include random numbers in questions, go to the Web site.*

7. THINKS (Tutorial Help In Numeracy Key Skill)

THINKS was developed at the University of Northumbria at Newcastle (UNN) as part of the TLTP3 project Capabil-IT-y. The first version was written to cover the content of the Quality Curriculum Authority (QCA) key skill *Application of Number*. It consists of fifteen tutorials – fractions and decimals, percentages, ratio and proportion, measures, estimation and approximation, area and volume, averages, tables, graphs, standard form, expressions, simple equations, formulae, Pythagoras' Theorem and trigonometry.

Authored in Question Mark for Windows, THINKS is intended for formative assessment – hence the assessments are termed tutorials rather than tests. Five question types are used: multiple choice, multiple response, selection, numeric and 'hot spot'. Each tutorial consists of an optional revision section and approximately 12 questions. After every question students receive immediate feedback which is specific to the response chosen, suggesting a correct response where an error may have occurred and/or a worked solution. End of test feedback refers students to other software titles, eg CALMAT Level 1, Mathwise and Transmath for further practice.

THINKS was piloted at UNN with groups of students for the departments of Law, Modern Languages, Economics and Chemistry. Students found the software easy to navigate, interesting and enjoyable to use, and a useful aid to learning. Pre and post-testing showed a good improvement in their understanding of the topics covered, and it was pleasing to note a general improvement in student attitude towards mathematics. QM Computing made THINKS available from their online question bank*, and this led to considerable interest in the software which is now in use at a number of FE and HE institutions in the UK.

Comments from academic staff suggested that the needs of students varied greatly – students from non-numerate disciplines required a greater emphasis on basic numeracy, while students from semi-numerate disciplines required further practice in algebra. To address this a number of new tutorials have been developed. The single fractions, decimals and percentages tutorial has been re-written as three separate tutorials, offering more practice for each topic. New tutorials include metric measures, directed numbers, order of operations, expanding brackets, simple factorisation, factorising quadratic expressions, solving quadratic equations, power notation, straight line graphs and the general equation of a straight line.

Improvements to the revision sections have also been made, and some revision sections have been completely re-written. To give a more 'mature' feel the multimedia calls to sound have been removed, along with many of the graphics. A new feature is that low scoring students are offered the opportunity to review the revision section at two points in every tutorial.

Conversion of THINKS to QM Perception is currently underway at Plymouth University. It is hoped to link between THINKS and the Mathletics question bank at Portsmouth for common topics, allowing students further practice as required. For further information on the recent developments to THINKS please contact Alan Heslington, Quality Enhancement Unit, University of Northumbria, Newcastle, NE1 8ST, email alan.heslington@unn.ac.uk

8. DIAGNOSYS

DIAGNOSYS is an expert-system testing shell for maths and other subjects, widely used in UK universities and abroad. DIAGNOSYS v3.3 now has a SELF-TESTING mode, which permits students (probably after taking the test initially) to select an area for improvement, then to make multiple attempts with the aid of Hints to try to brush-up that area, before taking a post-test. It also includes a number of minor improvements to the main testing shell and the test editor (renamed DIAGEDIT) and a full test design manual. MATHINPUT is a general-purpose maths input tool used by DIAGNOSYS and by other CAL developers. DIAGNOSYS v3.3, MATHINPUT v1.4 and DIAGEDIT v3.3 are available from the DIAGNOSYS web site.*

* links to all these Web sites are included in the Web version of this report at http://www.bham.ac.uk/msor/workshops/maths-caa

9. Wiley eGrade - announcement from John Wiley & Sons

Developed by Professor John Orr and the faculty and graduate students at the University of Nebraska-Lincoln, eGrade is web-based software that can process a wide range of math-based questions, while supporting content across all academic disciplines. It has been class-tested for more than two years at many different US universities. With many state-of-the-art features and functions that make it easy to use, eGrade is an instructional tool for homework management and assessment that works in many environments, including self-paced, monitoring and formal grading.

eGrade handles mathematical notation and technical subject matter. Students can use a preview button which converts keyboard input into typeset math notation and lecturers can use LaTeX input for new problems and exercises. Many learning models are supported, such as immediate, automated grading, comments on incorrect student responses, viewing of old tests to monitor progress, or adaptive mastery testing (customising each student's session based on his or her demonstrated learning needs), while repeatable testing allows for multiple tries and self-paced drilling of example problems. Each version of a test is different, and there are security features to ensure student identity for tests and exams.

Question styles available

- Formula
- Click-on-image
- Multiple Choice
- Numeric
- Short Phrase Essay
- Numeric with Physical Units
- Numeric with Margin of Error
- Matching
- Fill-in-the-Blank
- Choose all that apply
- Matrices

Experience in Authoring Questions in AIM

Dr D Lawson (d.lawson@coventry.ac.uk), Ms E Stone (ems3@coventry.ac.uk) and Dr J Tabor (j.tabor@coventry.ac.uk), Coventry University

Shortly after the LTSN meeting on interactive assessment in Birmingham, Theodore Kolokolnikov installed an AIM (Alice Interactive Mathematics) server on one of the author's desktop PC, by downloading a free java development kit and webserver and the AIM system itself. A less experienced person might have more difficulty in concluding a successful installation. From this PC, firewalls permitting, the system would be available globally for students to take tests, and for teachers to set tests. The system has subsequently been extensively tested by a second year student (ES) to explore the strengths and weaknesses of mathematical question setting, particularly compared to Question Mark Designer.

The purpose of this note is to provide those who didn't attend the meeting with an overview of the system and also to provide those who did attend with a summary of a slightly more in depth experience than was possible at the meeting.

As with all IT systems, the basic questions about interactive assessment are "Will I be able to use and modify material authored today in ten years time?" and "How easy is it to author material?". Questions more closely related to interactive assessment are "Is it easy to incorporate questions authored by others?", "Will students find the system transparent to use?", "Will it be easy to register students and process results?" and finally "Can I set a level 2 maths methods examination that is at least as good as a paper based examination with this system?". Read on for some of the answers with respect to AIM.

The AIM System

Alice is a wider educational project funded by the Belgian government and AIM is based at the Royal University of Ghent. It has to be emphasised that AIM is a work in progress. The system in use was at the stage of six months into a two year project. Some of the features criticised were discussed with Kolokolnikov, and these features may be in development at this time.

The AIM system comprises a kernel program written in java running along with Maple on an NT Web server.

The fact that Maple is only required on the web server is a major advantage. A student taking a test does not need to be able to access Maple locally. So the AIM system can be used by institutions which do not have a Maple site licence.

The AIM main screen is in HTML, and is therefore easily customised. Students navigate to their course, choose a quiz and access questions via a browser, which is quite standard now. Less conventionally, questions are authored using a browser. There is a hierarchy of 'courses' (individually password protected for access by different teachers), 'quizzes' and 'question files'. A quiz can be randomly generated from multiple question files to match sophisticated criteria. The authoring interface is essentially command line but very simple and very flexible. It is easy to learn how to set questions by looking at and copying example questions. Good author level documentation exists.

The main strength of this system is the close link to Maple. The question format allows the full power of Maple to be used not just in generating randomised questions and checking answers, but even in generating randomised graphics. Maple graphical output can be converted to a bitmap on the fly, and displayed over the Web in the standard way. It is even possible to use animated graphics from Maple, but probably not advisable since the download takes a long time.

Question/answer types can be text, number, function (of any number of variables, any variable names – Maple can handle it), vector or matrix. Vector and matrix entries can be functions etc.. It is also possible to use multiple choice and multiple selection questions where the alternatives are pure ASCII text. There is currently no hotspot question type, and it is not possible to organise questions to award partial credit.

Output to the screen can be based on the full range of HTML, including bitmaps, or Maple character based 'pretty print'. Super- and sub-scripts are possible in HTML, and support for Greek letters has been added, but 'pretty print' output is rather a jolt now, even though it is perfectly clear to understand.

If properly typeset mathematics is required, recourse to a bitmap may be necessary. This rather sidesteps the strength of the system, since such bitmaps cannot be sensibly and economically randomised. Bitmaps do appear to be a simple way to implement multiple choice questions containing mathematics, particularly if the surd symbol is required.

Currently, it appears that bitmaps for all courses must be located in a directory below the web server software rather than with the question files. Also, no way was found to upload bitmaps over the Web interface. These are very serious deficiencies that will severely limit possibilities for using the AIM system.

Student typed input must be in strict Maple syntax. For simple questions this is a small handicap, but the Maple syntax for calculus operations is very non-intuitive. It is possible to have the text input displayed as character 'pretty print'. Using strict Maple syntax is understandable at this stage of development, but developments to accept sloppy text syntax and convert to strict Maple syntax or some form of interactive maths input are likely to be necessary. It is possible to specify forbidden text in an answer. If the inverse of a matrix were asked for, then the Maple command inverse(...) with the original matrix as argument would match the answer, so it would be necessary to forbid inverse in the input.

Once a test is completed, the answers can be 'validated' or 'marked'. It would be natural to assume validation would merely check answers for type and syntax and display the 'pretty print' version, however the validate button does also check for correct answers. This is a problem since, if a wrong answer is given, retries are allowed but an author defined penalty is subtracted for each wrong trial. It would be beneficial to have a true validate button for each question. Quizzes can have a final date. After this date, questions can be marked and solutions are displayed if they have been provided by the author. Both validation and marking test for syntax and answer type, e.g. no marks for a wrong trial would be deducted if a number was input when a function of x was expected.

To author questions in a question file or to construct quizzes, password protected 'teacher' level access is required to a course, but these tasks can be done using a remote browser rather than needing direct access to the server. The author interface for setting questions is essentially text command line and requires a little knowledge of HTML tags and good knowledge of Maple syntax. The close coupling with Maple, where a question may contain several lines of Maple statements, more or less dictates this form of input. There is also the great advantage that if you have access to example questions, it is immediately apparent how to construct similar questions. Anyone who is already a Maple user will feel very much at home with this means of setting questions. A striking feature is the economy and speed with which quite sophisticated questions involving randomisation and graphics can be set.

Student registration and results reporting facilities are provided, but were not properly tested for this note. The major effort was directed to exploring how to set questions.

Conclusions

AIM is not yet a mature and well tested system for interactive assessment in mathematics. This should not be expected so early in the development cycle. The question is whether it has the potential, over the course of the planned project, to become sufficiently flexible and widely accepted to repay the investment of time needed to beta test and develop 'production' grade assessment material.

The use of standard browser technology and the strong links to Maple make the AIM kernel very lean. Many improvements will come automatically through browser and Maple upgrades allowing the AIM development to be focused on specific assessment issues. This should allow for portability and longevity. It certainly seems clear that this paradigm for interactive assessment in mathematics will be around for a long time.

For those authors familiar with Maple, question setting is as easy as typesetting paper based questions and the full power of Maple is available for randomisation, including graphics etc. It is also entirely possible to incorporate questions set by other authors. However (and this is a good feature in our opinion) it is not possible to lift other authors' questions with just student level access to their question bases. Either the question files themselves or password protected 'teacher' level access is needed.

Currently, the student interface does need work (and will doubtless receive it). The character based mathematics output is now less acceptable than a few years ago. However if browsers and Maple adopt XML as an input/output standard this problem evaporates with no further work on the AIM system. Compliance with Maple input syntax does restrict the number of students who will be comfortable with the system. More development is needed here. Also the mark/validate system where students can check correct syntax needs to be more transparent.

It is hard to comment on how well the student registration and results reporting systems in AIM will work when faced with large numbers of students. It would be expected that, in line with the lean and mean philosophy of the system, standard input and output such as comma separated variable spreadsheet data will be provided and easy to use.

Is it possible to set a level 2 exam paper? Not yet, since there is no means of awarding partial credit. However the link to Maple means that class of questions that can be easily set and reliably marked is unboundedly rich. It is surely just a matter of a little development to the AIM system to allow for partial credit in a way that would be both very close to a human marker and relatively simple for the question author.

While it is clear that the AIM system is in need of a great deal of development, this is to be expected in a system less than a year along the track. The surprise is that AIM has got this far so quickly. The basic paradigm of the system is so powerful, elegant and adaptable and the current author interface so easy to use, at least for a limited class of questions, it is manifest that AIM is worth serious attention from anyone interested in interactive assessment in mathematics.

Assessment over the Web using CUE

Dickon H Jackson (dickon.jackson@hw.ac.uk), David J Fiddes (d.j.fiddes@hw.ac.uk) and Cliff E Beevers (c.e.beevers@hw.ac.uk), Heriot-Watt University

This article consolidates a talk presented in the Department of Mathematics at Birmingham University on May 24th 2000 at a meeting of the LTSN Mathematics Centre. It provides details on the new assessment system known as CUE, currently being used at Heriot-Watt University to deliver a range of tests in the SCHOLAR distance learning project. CUE has developed from earlier assessment engines used to service tests in the CALM Project for Computer Aided Learning in Mathematics, and assessments in the Mathwise Project. The article gives some background to this development which is based strongly on the commercial testing package Interactive PastPapers [1] but which now operates over the web.

In an article in the last newsletter, Beevers et al [2] described the use of the computer in mathematical assessment over the last fifteen years on the Heriot-Watt University campus. Interactive assessment at Heriot-Watt University as practised by the CALM Project for Computer Aided Learning in Mathematics takes four forms:

1. There is a diagnostic multiple choice test at the start of year 1 for Service Mathematics students (see www.icbl.hw.ac.uk/marble/maths/public for details);

2. There has been a Monitoring test for CALM Calculus units each week for 400 students per year throughout the 1990s (see www.calm.hw.ac.uk for more information);

3. Self-testing is a feature of the CALM pre-university Mathematics delivered into schools by CD; and

4. A Grading test has been administered providing 40% assessed by computer for 250 students each year since 1994.

From the early days the features in CALM tests included randoms in questions, random choice of questions, structured response answer types deliverd in three modes of test to cater for a wide spectrum of student ability, rapid feedback on predictable wrong answers and a results service for students and teachers.

Following a generous prize award of £30k to the CALM Project team in 1995 by the Bank of Scotland, the commercial product Interactive PastPapers was created. This package delivers mathematical questions from core A-Level and Scottish Higher Mathematics with Random parameters in most questions. An Input tool was used to aid the entry of mathematical answers [3]. It was also possible to submit multiple entries — vectors, factors, co-ordinates etc and special answers like those using scientific notation or requiring significant figures. Educational features like partial credit, key parts and differential marking are included in the package; these are further described below. In addition, it is possible to have multiple saved tests, there is on-line context sensitive help and flexible methods for question selection. A demo is available at www.calm.hw.ac.uk.

Issues of Partial Credit

It has been the philosophy of the CALM Group to listen to student reaction and to build educational findings into the software. To this end, the problem of partial credit has been one that has been difficult to resolve. Questions naturally break down into a series of parts, which require answers to complete the question. Proficient students have little problem in responding to this requirement but to assist those having difficulties, the notion of steps leading to a key part was devised. To illustrate the use of key steps, first pioneered in the Mathwise materials [4], consider the example:

Question
Find the tangent to the curve given by $f(x) = x^3+4$ at the point where $x = 1$

This question can be set with one key part which asks:

1. Equation of tangent is y = ? 3x+2

Here, three marks can be given but there is no opportunity for partial credit for the student who does not know where to start. Whereas in Mathwise and Interactive PastPapers *if steps have been included* then a student has the option to ask for further steps such as in this case:

1.1 Derivative of f with respect to x is ? $3x^2$

1.2 Gradient at x = 1 is ? 3

Again, three marks can be on offer with steps 1.1 and 1.2 each worth 0.5 marks and then at least partial credit is possible. This allows more students to progress as they learn in self-testing style and this device allows partial credit too during a grading test.

There are other ways of delivering automatic partial credit when, for example, an answer is almost right but not in the appropriate form. Two examples suffice: in the first an answer of one half might be required to be given as a fraction 1/2 so if the user types 0.5 then partial credit can be awarded and the message "Give answer as a fraction — 50% partial credit" can be issued.

In a second example, if the answer to a length of a vector is sqrt 2 say and the student types value 1.414 then both Mathwise and Interactive PastPapers can offer partial credit with the warning "Your answer should contain a square root — 75% partial credit".

The CUE Assessment System

Over the last couple of years, the CALM Group has formed strategic alliances with a leading UK Examination Board (University of Cambridge Local Examination Syndicate, UCLES) and a local software house in West Lothian (EQL). Our technology has migrated to provide both PC and web-delivered tests with many of the

features of our earlier products. The basic CUE software suite comprises a Question Editor, a Test Editor and the Assessment Engine. The Assessment Engine exists as both PC and web applications, allowing the use of exactly the same questions in either mode of delivery. Focus is given to the web-based Assessment Engine in this instance. A demonstration of the system illustrated the capability for rapid local authoring of test materials with subsequent submission to, and operation on, a remote web server.

Authoring Tools

The Question Editor is a major advancement in terms of ease of content authoring and reductions in preparation time. Questions are saved as XML files, with content MathML being utilised in the storage of mathematical expressions [5]. This gives a reusable format with greater opportunity for conversion and transfer to other assessment systems. Similarly, the Test Editor gives the tutor a quick and simple solution for compiling questions into a test. Included in this are the settings for a standard range of feedback features. This allows the test to be defined in some level of practice mode, where the student can monitor their own progress, or full exam mode where no active feedback is given to the students as they answer each question.

Assessment Engine

The Assessment Engine renders the questions and handles the student responses on a part-by-part basis. Communication complications with the server will therefore, at most, result in the loss of a student response for a single part and not an entire question or test. Furthermore, student results are stored incrementally thus allowing an accurate profile of student performance within the test to be ascertained. This has further benefits should the student question the marks they have been awarded. Figure 1 gives an example of a three-part question running in the CUE Web Assessment Engine. Buttons titled *Steps* indicate that sub-parts are available for that particular part.

Part Types

CUE supports an expanding range of part types. However, being built upon the basis of the standard CALM/ Mathwise Judged Mathematical Expression, the system is capable of handling complex mathematical expressions as well as standard numerical answers and includes generation of random numbers from user defined sets. In addition to this are several other part types,

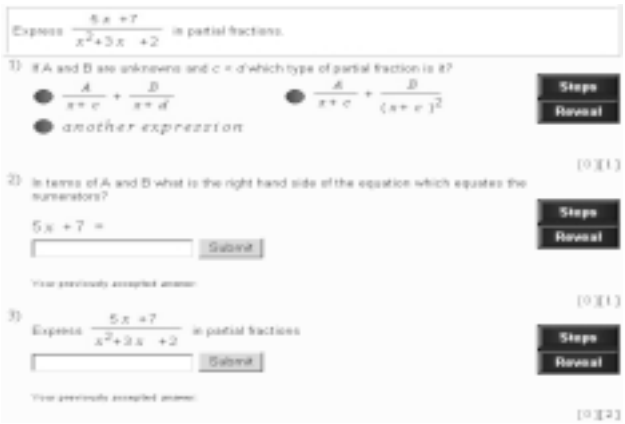

Figure 1: Example question in CUE Web Assessment Engine with feedback active.

summarised below:

- Multiple Choice – Allows different display mechanisms.
- Multiple Response – Allows user defined marking scheme.
- Word/Phrase Match – Regular expressions.
- Hotspot – Multiple hot spot definition and user defined marking scheme.
- Essay – Open answer with offline marking.

These broad answer types are merely illustrative of a wide range of variants now available in the CUE system. The Question Editor allows the author to build questions with any mixture and number of these part types as key parts or steps.

Conclusions

The CUE assessment system has been chosen to provide a range of tests for the distance delivery of Mathematics, Physics, Chemistry, Biology, Computing and Engineering materials into Scottish schools in the SCHOLAR Project. SCHOLAR is aimed at the new Advanced Higher qualification offered by the Scottish Qualifications Authority and is similar to an A-Level programme in the rest of the UK. Over the next few months it is expected that the experience of delivering such an ambitious programme will provide further improvements to the CUE system.

References

[1] C E Beevers, D J Fiddes, G R McGuire, D G Wild and M A Youngson, *Interactive PastPapers for Higher and A-Level Mathematics*, Lander Educational Software, Glasgow, 1997

[2] C E Beevers, D J Fiddes, D H Jackson, G R McGuire and M A Youngson, *Computer Aided Assessment in Mathematics at Heriot-Watt University*, Maths, Stats and OR Newsletter, February, 2000, p17 - 19,
[3] C E Beevers, D J Fiddes, G R McGuire, M A Youngson, *The Emerging Philosophy behind Computer-Based Assessment*, Teaching Mathematics and its Applications, 1999, Vol. 18, No. 4, 147-149
[4] C E Beevers, P Bishop and D Quinney, Mathwise Diagnostic Testing and Assessment, Information Services and Use, 18, IOS Press ISSN 0167-5265, 1998, 191 - 205
[5] D H Jackson, D G Wild, *The Use of XML and MathML for Interactive Non-Multiple-Choice Assessment*, Twelfth International Conference on Technology and Collegiate Mathematics, San Francisco, 1999

Computer Test Setting using TAL

M D J Barry (mike.barry@bristol.ac.uk), J H Sims Williams (jon.sims.williams@bristol.ac.uk) and R E Wilson (re.wilson@bristol.ac.uk), University of Bristol

The Test and Learn or TAL system at the University of Bristol provides multiple-choice computer-based tests on first year engineering mathematics. The tests are now available over the Internet. This system, based upon a large structured database of questions, has since been expanded into progress testing in other subjects, notably chemistry, and the database is being extended to include other point and click type responses. It is now available with full colour graphics facilities from a WEB-based server. The workshop session at Birmingham was aimed at guiding lecturers in specifying tests that are suitable for their own students. The structure permits one to be specific as to the material included in or excluded by such tests, and hopefully ensures that the tests and the question descriptors such as question type, difficulty, time to do, and suitability for a given student group, meet the lecturer's need. In time it is intended to create a database large enough to generate a sufficient number of different tests as to allow testing in a totally open environment.

Question Descriptors

There are 1253 first year engineering mathematics questions in the database, and these are broken down into various Main Topics; for example, 229 involve Single Variable Differentiation. Each Main Topic is decomposed into subtopics, for example: Simple Algebraic Functions, (10); Chain, Product or Ratio Rule, (15); Function of a Function, (19); Parameters, (8); Maximum, Minimum, Stationary Point, (15); Higher Derivative, (12); Practical Application, (25); and Simple Model Construction, (27): the numbers of questions are shown in brackets. A priority order operates so that the test setter knows that Simple Algebraic Functions only includes powers, polynomials and simple surds but should a higher skill such as function of a function be involved then this predominates. For example, a Simple Model Construction might involve the student in an elementary construction such as finding the total surface area of a rectangular box whose linear dimensions are known.

Questions are classified by a tree structure with mathematics as the root of the tree. So a question like:

What is the angle, in radians, between the vector ***p*** *= [2,-5,-3] and the x-axis?*

is classified in the tree as:

Mathematics/vector algebra/direction cosines/ evaluation.

'Evaluation' specifies that a numerical value is required.

There are further descriptors such as style, theoretical versus practical content, and suitability for a given degree programme. Of most immediate concern however are the Time to Do and the Difficulty. Initially these were allocated values by the database designers but repeated usage now allows empirical values to be used, which are based upon the actual time taken by students to respond to a question and the percentage of correct responses to it. For example, we ascribe 'Time to Do' as the mean time taken by students to get the question correct plus one standard deviation. The difficulty or facility of a question is measured on a 1% to 99% scale, where this represents the percentage of students in the particular group who can get the question correct.

Setting a Test

By logging in at 'http://www.tal.bris.ac.uk/Login.htm' as a Lecturer, using the appropriate password, a test setter can select from the menu to set a test, specifying and excluding categories and individual questions as necessary. He or she is also empowered to specify the time and difficulty of the test as a whole within the parameters that might be stipulated. A typical specification for a 7-question test might be as follows:

Qn No	SubTopic	Number Available
1	Simple Algebraic Function / Chain Rule	25
2	Function of Function	19
3	Practical Application	25
4	Simple Model Construction	27
5	Max/Min/ Stationary Pt	15
6	Higher Derivatives / Parameters	30
7	Higher Derivatives / Parameters	30

Time to Do: 32.1 ± 5 minutes
Average Facility: 60 ± 3%
Good Tests: 14

The setter would thus have 14 tests, rather than a potential maximum of 15, with a mean execution time of just over 30 minutes with the facility as shown. Note that the same topic is used for the last two questions and so different questions with the same classification are chosen. The setter, knowing that questions are presented to the students on the screen in random order, as are the multiple choice alternatives, might feel that the number of tests so set can be presented to students in an open environment. In other words each student can take the test during a period, of say four days, at a time and place of their own choosing.

Security is a key issue in open testing and is very difficult to enforce totally. One thought has been to present a scanned photo of the student under test in the corner of the screen, though this has yet to be implemented. In any case the purpose of testing in first year engineering mathematics is for formative assessment with the bulk of the summative marks going to a traditional written examination. It is put to students that the TAL tests are simply to 'test and learn' and should be honoured as such, with a clear explanation to the effect that an infringement constitutes an offence under university examination regulations.

Collaboration

The strength of the TAL system at Bristol is that the large and versatile database is proving to be of inestimable value in the objective and efficient assessment of large numbers of students. They respect the objectiveness as this eliminates the natural variation that always exists between individual markers of large common purpose examinations. Many would also like to use TAL on an individual basis to benchmark their own learning and understanding; this would require considerable database enlargement. It is very much in the interest of hard-pressed institutions to collaborate in such an enlargement and this must be the way forward.