Limitations of Authoring Tools for the Production of Interactive Mathematical and Graphical Software

Heather Mays and Binh Pham
School of Information Technology and Mathematical Sciences.
University of Ballarat. PO Box 663, Ballarat. VIC. 3353.
hjm@ballarat.edu.au , binh@ballarat.edu.au

Abstract

In this paper we outline current approaches to Computer Aided Learning (CAL) in Mathematics and Graphics. Due to the limitations of the current technology, most of the software that has been produced is essentially hypertext with some multimedia additions. The creation of an environment suitable for the production of dynamic, interactive teaching software for Mathematics and Graphics requires the use of an authoring tool as well as an interpretive tool which is capable of parsing text input and producing symbolic, numeric or graphical output. Some specific suggestions for future developments in mathematical and graphical CAL are outlined. This is the basis of an ongoing research project at the University of Ballarat.

Keywords

mathematics, graphics, CAL, dynamic, interactive

1. Background

Tools currently available for authoring interactive software are of the 'cut-and-paste' variety. That is, they are not capable of producing truly dynamic software for the teaching of mathematical and graphical techniques. Instead, the presentation screens are 'dead' output that have been imported into the authoring software. Whilst most authoring tools have some low-level mathematical functions built in and the end-user has some control over parameters that govern the output on the screen, the author must have already created the output elsewhere. This is the reason that nearly all the projects aimed at teaching mathematics in an 'interactive' environment are really hypertext with some multimedia additions (animations, sound and video). Examples from the United Kingdom include the Renaissance project, the CALM project and the Mathwise project (see section 3).

Whilst authoring packages can call up other applications such as computer algebra systems or graphing software, they are capable of only limited data exchange with such tools and typically the exchanged data can only be text. Underlying this problem is the notation used in both mathematics and graphics. The usual way of producing mathematical equations on screen is to use bitmaps that have been imported from another source (e.g. Equation Editor in Windows). When the bitmap is imported, the symbolic essence of the equation is lost.

Computer algebra systems (e.g. MAPLE and Mathematica) are capable of parsing text input and producing symbolic, numeric or graphical output. However these packages are interpretive, ie. they cannot be used to generate compiled code that can run external to the package used to generate it. To create an environment suitable for the production of dynamic, interactive teaching software for Mathematics and Graphics, integration of the above two types of software is needed. In addition, good self-taught and self-evaluated courseware should offer the facility for students to generate their own solutions in text and / or graphics by writing and executing their own programs. It would also be advantageous to allow students to share ideas and compare solutions. This paper analyses the problems mentioned above and discusses some of the options available for their solution.

2.CAL in Higher Education

The use of Computer Aided Learning (CAL) as an adjunct to more traditional modes of course delivery in tertiary education has grown in the last five years (McDonough, Strivens and Rada, 1994a). There are a number of reasons for this:

- increased demand for remedial / bridging courses;
- increased enrolments in tertiary education (DEET reports, 1994);
- higher teacher / student ratios;
- students entering tertiary education with poor background knowledge (Gladwin, Margerison and Walker, 1992);
- greater provision of distance education (DEET reports, 1994); and
- the move to ‘integrated’ courses.

However, there still remain a number of impediments to the uptake of CAL (Darby, 1992; McDonough, Strivens and Rada, 1994a, 1994b; Hammond *et al.* (1992); Gladwin, Margerison and Walker, 1992). They include:

- shortage of lecturers’ time for creation and evaluation of materials;
- poor rewards for teaching innovation or courseware development;
- lack of support staff, training and information on suitable materials; and
- the ‘not invented here’ syndrome.

Because Mathematics is fundamental to all Science and Engineering courses, it is essential that students in such courses have a solid grounding in Mathematics. Gladwin, Margerison and Walker (1992) identified poor mathematical background of students enrolled in chemistry courses as an important factor in their poor performance. Bacon (1992) reported similar results for students enrolled in Physics courses. In Australia, DEET profiles for the period 1992-1994 indicate that while enrolments in Engineering and Science courses increased over the period, enrolments in and funding for Mathematics and Computing units have declined. Partly this has been due to the move towards ‘integrated’ curriculum combined with a decline in secondary school enrolments in Mathematics (Yearwood and Glover, 1993).

CAL has long been recognised as having two major benefits—allowing students to work anonymously at their own pace and relieving teachers of much of the repetitious and unsatisfying work involved in the students’ acquisition of low-level skills. The case for the use of CAL in tertiary mathematics is further strengthened by its ability to aid students in visualisation, to provide an environment that encourages exploration and to enliven the curriculum. Computer Algebra Systems (CASs) have become very popular in the last few years as an adjunct to teaching (Yearwood and Glover, 1993). Apart from their symbolic and numeric capabilities, CASs such as MAPLE are capable of plotting in two or three dimensions as well as producing animations of output. However these systems are interpretive. Full exploitation of the capabilities of such tools requires the creation of an environment that combines an authoring tool and a mathematics and graphics package.

3. Current Projects in Mathematical CAL

A number of researchers have been working in the area of mathematical CAL in the last ten years. Some of the major projects from Australia and the United Kingdom are now considered.

3.1 United Kingdom

The United Kingdom has shown the greatest activity in the area of CAL in Mathematics education over the last ten years. This is due largely to two initiatives—the Computers in Teaching Initiative (CTI) and the Teaching and Learning Technology Programme (TLTP). The principal aim of the CTI was to collate, organise and disseminate information on the use of computers in teaching. In February 1992 the Universities Funding Council (UFC) launched the first phase of the Teaching and Learning Technology Programme. The aim of the TLTP was to improve the productivity and efficiency of teaching and learning through the use of modern technology.

3.1.1 The CALM Project

The CALM (Computer Aided Learning in Mathematics) project started at the Heriot-Watt University in 1985 with funding through the CTI. The aim was to use a computer-based system to replace the traditional, pen-and-paper Calculus tutorials for first-year Science and Engineering students. This system was designed to supplement rather than replace the lecture system as the traditional tutorials had been poorly attended and the results for first-year Mathematics generally were considered to be less than satisfactory.

The first phase of CALM consisted of units in introductory differentiation and integration, numerical analysis and introductory differential equations. All units had a common structure of a theory section, a section of worked examples (including games and mathematical modelling aimed at motivating the students) and a test section. At the time that CALM started, it was felt that no satisfactory authoring package existed and so the entire system was coded using ProPascal. As the project proceeded, a library of functions and graphical routines was created. CALM provides all the usual CAL requirements, viz. displays of information, animations, ability to control the path and the rate of progression through the package, ability to accept user input, ability to send messages or provide on-line help to the end-users and some management facility to log student progress.

The intention of the CALM team was to produce a package that allowed the user to input mathematics in a manner which mimicked the pen and paper approach. This forced the team to address one of the most important problems with mathematical CAL—the display and input of mathematical symbols. This was achieved by using the extended ASCII character set, redefining some character codes and using the F1-F4 function keys for four of the most commonly used symbols (viz. $\int$, $\surd$, π and ^{2}). Detailed instructions on the input of mathematical expressions were provided as hard-copy to users. CALM only allowed for one-line input but the authors felt that this was advantageous as it forced students to think about the meaning of their input. Because of the variety of ways that a correct answer can be represented in mathematics, the team also needed to create a method for evaluating algebraic expressions and to compare a student's answer with the correct answer. The evaluation of functions was limited to real-valued functions.

The test section allowed the students to nominate the level at which they wished to work. Depending upon the level nominated, the package provided some guidance on the answering of questions. The worked examples section included animations, mathematical games and some modelling exercises.

The purpose of this section was to supplement the theory and to provide motivational examples to the students.

The second phase of the CALM project is currently under way. The content includes powers, trigonometry, analytic geometry, vectors, differentiation, integration and elementary statistics. The Project now makes use of the Authorware language rather than programming in ProPascal. The theory section is now delivered in both detailed and summary modes. New features include a new section in each topic of self-assessment questions, linking between appropriate theory, worked examples, enhanced graphical displays, more animated sequences and an overview of each topic.

A unit test completes each unit of courseware. These units are being trialled in a number of schools and initial comments indicate that they will provide an additional tool for school teachers.

3.1.2 The Renaissance Project

The work by Robert Harding, Douglas Quinney and others began with the production of Computer Illustrated Texts (CITs) in 1985 (Harding and Quinney, 1990). These were a combination of text and software which combined the 'browsability' of a text book with the dynamics of software. The final learning environment enabled greater participation by the reader than a standard textbook. Because the software packages were able to accept user input, users could control the values of parameters and produce dynamic illustrations of mathematical concepts. The topics covered included Fourier series and transforms, introduction to probability and statistics, number theory, introduction to group theory and electric circuit theory. The software packages were written in Pascal and included both mathematical and graphical routines.

The Renaissance project developed so that the CITs could be converted to hypertext using HyperCard. The first phase involved the creation of templates and the second phase was aimed at creating multimedia mathematics modules in algebra, coordinate geometry, calculus and mathematical modelling. Whilst the team found that the HyperCard environment made manipulation of text and bitmap graphics and hence the creation of a friendly user interface very easy, it was not well-suited to handling mathematical expressions and contained no mathematical or graphing tools. It also enabled only limited data exchange with other applications (such as computer algebra systems).

These difficulties led to the following comment (Harding and Quinney, 1995, p. 7):

> There is perhaps a danger that the technical tail is wagging the educational dog, but today all over the world, education is not able to direct technological development. Educators have to make use of platforms which have been developed for the business market, although one should not underestimate the importance of the education market to manufacturers.

To overcome these difficulties, the Renaissance team, in conjunction with Michael Beilby from the Mathwise project, is currently working on its own mathematical toolkit for incorporation with HyperCard. The purpose of the toolkit is to overcome the same problems encountered by the CALM team, i.e., the inability of the current authoring languages to easily accept and evaluate mathematical input from the user. A similar toolkit is being built to support the Toolbook environment. These toolkits represent progress towards the creation of an environment which overcomes the usual problems encountered by the authors of mathematical courseware. The aim is to extend the design to produce a toolkit which is genuinely cross-platform (ie. independent of both the software and the hardware being used by the author).

3.1.3 The Mathwise Project

Mathwise is a computer-based learning environment for Mathematics and is another project which was funded by the TLTP (Beilby, 1993). Related to this project is the construction of the Assessor program (Beilby, 1994). The Mathwise files have been built using authoring packages such as Toolbook and Authorware Professional, as well as other specialised packages including Word, Windows Help, a graphical package, a calculator, a test package and a MAPLE processor. Because of this, indexing and cross-referencing of files is fundamental to Mathwise.

The Mathwise project has already produced over 30 modules including material covering introductory calculus, mathematical logic, vector algebra and trigonometrical functions. The purpose of using an 'environment' with an indexing system rather than a single training package is to enable and encourage students to study in a more natural way. That is, students can study the information provided in the modules as well as browse through the system and employ other resources such as the graphing tool or the MAPLE processor.

Although there are a number of testing packages available commercially, none of these is particularly suitable for the testing of Mathematics. The aim of the Assessor project was to produce a computing environment which could handle the format of Mathematics questions and to identify mathematically equivalent expressions. The Assessor program is built around a core of dynamic link libraries (DLLs) produced by Michael Beilby and conforms to the standards set by the Mathwise project. These DLLs are in two parts - one for the display of the questions and the other is for student input of answers. The latter is a menu-driven expression editor which is similar to Microsoft Equation Editor. This editor collects input which is to be evaluated and compared with the correct answer. It displays the expressions in the normal way that they would be hand-written using a combination of conventions from symbolic manipulators and the TEX language. The current form of the program includes a database of question files which are indexed by topic.

3.2 Australia

In Australia, there has been one very significant project viz. the Computer Assisted Learning and Testing System (CALTS) created by Ernest Yu at RMIT (Yu, 1990; Yu, 1994).

3.2.1 The CALTS project

The development of the CALTS system commenced in 1988 to create a Mathematics testing system for use by prospective students. At the time, no suitable authoring language existed which was both easy-to-use and capable of displaying graphical images and so dBase was used to create CALTS. Like the products from the United Kingdom, CALTS provides an authoring system, a testing / reporting system and a management system. CALTS is capable of all the same functions that the first phase of the CALM project can achieve but has gone some way further with the level of mathematics that it can handle including some complex arithmetic, linear algebra, numerical integration, numerical solutions to equations and differential equations and some vector analysis. The authoring system is capable of displaying mathematical input, interpreting algebraic expressions, producing colour graphics and to externally compile programs for lessons or tests.

During the period 1993-1994, the CALTS system was modified so that it can now provide students with individualised assignments, allow for electronic submission of assignments with a specially developed encoding algorithm, provide immediate feedback and allow students to regularly view their performance records. Because CALTS was built using a database approach rather than the normal hypertext approach, it is easier to store and update data. The network update facility provides added security for the system.

4. Problems with Mathematical CAL

Although Mathematics is a discipline that is well suited to an interactive, computer-based environment, it is probably the most difficult to incorporate into a CAL package. Many authors have discussed the problems encountered in authoring interactive mathematical software that arise because of the symbolic nature of Mathematics (Harding *et al*.,1995a and Beevers *et al*.,1992). Typical problems are fractions and the dynamic nature of the specialised character set required (for example, √, ∂ and ∫). The usual way of producing mathematical equations on screen is to use bitmaps that have been imported from another source. A major problem with mathematical CAL is the many ways of expressing algebraic constructions. For example, all the following are equivalent: $x^2 - 1$, (x-1)(x+1) , (x+1)(x-1), xx-1. A CAL package should be able to identify that these expressions are equivalent, and also needs to be portable, flexible and extendible with an object-oriented interface. It should also provide audio, video and animation capability.

Current CAL systems deal extensively with linear algebra and numerical methods which require symbolic and numeric capabilities. Very little effort has been expended on the teaching of mathematical and graphical modelling of dynamic phenomena such as kinematics and dynamics. Consequently, a, real-time, interactive 3D graphics capability should be integrated into the authoring environment. Students should be able to retrieve stored graphical displays and to write and execute their own computer programs to produce static or dynamic graphical output from within the CAL package. They should also be able to manipulate modelling parameters to study the effects within a reasonable timeframe.

Another problem is the static manner in which current CAL systems deal with student responses and their evaluations. Student responses are classified as correct or incorrect, without allowing for any slight deviation from the norm. There is a need for a more dynamic assessment scheme which caters for the idiosyncratic behaviour of individual users. This can be achieved by constructing a knowledge-based system with some capability of intelligent and fuzzy reasoning. This system may be extended to provide a learning capability which trains the system to recognise new valid responses. Once these valid responses are recognised, they would be added to the system. Such a system would require the ability to retrieve information and perform similarity matching on them. An example of a similar system is The Cardiac Tutor which teaches cardiac resuscitation procedures (Woolf and Hall, 1995).

It is also essential for students to be able to share ideas with others while learning because this generates enthusiasm and encourages discovery. Current CAL systems do not cater for this. A database would need to be created to record individual responses in order to monitor students' progress and to allow students to share knowledge. The control of whether certain information should be kept private or be made public is exerted through public or private links to the database.

Another issue worth mentioning is the efficiency of a CAL system that is capable of providing a real-time, fully interactive and dynamic multimedia environment. As the generation of real-time interactive 3D graphics and the retrieval of information from a knowledge-based system are resource-intensive, it is important to investigate efficient and effective techniques for performing these tasks and for determining the appropriate hardware platform. Furthermore, the hardware platform used has to be sufficiently powerful to allow real-time responses.

5. Conclusion and Future Directions

We have examined and analysed the limitations and requirements of mathematical and graphical Computer Aided Learning packages available in Australia and the United Kingdom. We are currently developing a computer-based learning environment which addresses these limitations. The environment will allow students to write and execute their own computer programs to produce symbolic, numeric and real-time 3D graphics output. Students will be able to store output in a database. Assessment will be monitored using a knowledge-based system in which fuzzy rules are constructed on the probability that a response is correct. This environment will better cater for individual differences and encourage an active and exploratory mode of learning.

6. Acknowledgments

The authors would like to thank Ms. Pat Mann, Planning Officer at the University of Ballarat, and Mr. Ernest Yu, Senior Lecturer from the Department of Mathematics at RMIT, for their help in compiling this paper.

7. References

Bacon, R. A. (1992). The use of computers in the teaching of physics, *Computers and Education,* Vol. 19, No. 1-2, pp. 57-66.

Beevers, C. E., Cherry, B. S. G., Foster, M. G. and McGuire, G. R. (1991). *Software tools for computer aided learning in mnathematics*, Avebury Technical, Aldershot, England.

Beevers, C. E., Foster, M. G., Korabinski, A. A. and McGuire, G. R. (1993). The use of computer aided learning (CAL) in mathematical education, *Collegiate Microcomputer*,Vol. XI, No. 3, pp. 199-204.

Beevers, C. E., Foster, M. G., McGuire, G. R. and Renshaw, J. H. (1992). Some problems of mathematical CAL, *Computers & Education*, Vol. 18, No. 1-3, pp. 119-125.

Beilby, M. (1993). Mathwise: A computer-based learning environment *Maths & Stats,* Vol. 4, No. 4, pp. 2-5.

Beilby, M. (1994). Making a feature of assessment, *Maths & Stats,* Vol. 5, No. 2, pp. 6-10.

Darby, J. (1992). The future of computers in teaching and learning, *Computers and Education,* Vol. 19, No. 1-2, pp. 193-197.

DEET (1992). *Higher Education Division Profiles 1992-1994. Agreed teaching profiles for higher education institutions in the Unified National System for the 1992-1994 triennium,* Australian Government Publishing Service. Canberra, Australia (ISBN 0-642 18063 6).

DEET (1994). *Higher Education Division. Selected higher education student statistics 1994,* Australian Government Publishing Service, Canberra, Australia (ISBN 0-644 42975 5).

Gladwin, R. P., Margerison, D. and Walker, S. M. (1992) Computer-assisted learning in chemistry, *Computers and Education,* Vol. 19, No. 1-2, pp. 17-25.

Hammond, N., Gardner, N., Heath, S., Kibby, M., Mayes, T., McAleese, R., Mullings, C. and Trapp, A. (1992). Blocks to the effective use of Information Technology in Higher Education, *Computers and Education,* Vol. 18, No. 1-3, pp. 155-162.

Harding, R. and Quinney, D. (1990). Computer Illustrated Texts (CITs) for teaching numerical analysis, *Computers & Education,* Vol. 15, No. 1-3, pp. 205-211.

Harding, R. D., Lay, S. W., Moule, H. and Quinney, D. A. (1995a). Multimedia interactive mathematics courseware: The mathematics experience within the Renaissance Project, *Computers & Education*, Vol. 24, No. 1, pp. 1-23.

Harding, R. D., Lay, S. W., Moule, H. and Quinney, D. A. (1995b). A mathematical toolkit for interactive hypertext courseware: Part of the mathematics experience within the Renaissance Project, *Computers & Education,* Vol. 24, No. 2, pp. 127-135.

McDonough, D., Strivens, J. and Rada, R. (1994a). Current development and use of computer-based teaching at the University of Liverpool, *Computers & Education,* Vol. 22, No. 4, pp. 335-343.

McDonough, D., Strivens, J. and Rada, R. (1994b). University courseware development: Comparative views of computer-based teaching by users and non-users, *Computers & Education,* Vol. 23, No. 3, pp. 211-220.

Woolf, B.P. and Hall W. (1995) Multimedia pedagogues: Interactive systems for teaching and learning, *Computer,* Vol. 28, No. 5, pp. 74-80.

Yearwood, J. L. and Glover, B. M. (1993). Computer algebra systems in tertiary engineering mathematics, *Working Paper 3/9,* School of Mathematics and Computing, University of Ballarat.

Yu, E. (1990). Building blocks of an intelligent tutoring system, *Proceedings of the International Conference on Information Technology,* Kuala Lumpar, Malaysia.

Yu, E. (1994). CALTS and computer managed assessments, *Proceedings of the CML Conference,* Melbourne, Australia.