Developing educational software for Design and Technology

Tony Hodgson

Introduction — the need for appropriate computer software

Information technology is here to stay. Despite the usual lack of resources, no teacher can escape the need to consider AT5 (or should that be AT3!) of the Technology National Curriculum. The non-statutory guidance for technology suggests that IT capability is best developed through other subjects, rather than being developed separately, and highlights the need for pupils to *develop* their IT skills only when they are ready to *make use* of them. Teachers of design and technology will identify with the ethos of this approach and so they are well placed to consider the progression of a pupil's IT development within their subject. The non-statutory guidance outlines five strands of IT capability which are often used as a starting point for any discussion related to the teaching of IT in schools:

> developing ideas and communicating information;
> handling information;
> modelling;
> measurement and control;
> applications and effects.
>
> Tracing these strands in the programmes of study and statements of attainment can provide a framework for ensuring progression. In secondary schools it can also help in deciding which subjects will teach the various aspects of IT capability. [1]

Using these strands to provide an overview of IT capability across the whole curriculum will provide a useful approach for some teachers, particularly IT co-ordinators. However, in order to identify how progression in IT capability will directly support progression in specific subjects, it is more useful to consider how IT will enhance teaching and learning in that subject. A detailed discussion of how the potential for IT in design and technology may be identified through subject based considerations can be found in the paper presented to IDATER '92 (Hodgson, 1992):

> The potential for IT in design and technology can be categorised through typical use... A range of categories might be:
>
> Graphic Communication
> Computer Aided Design and Manufacture
> Research/Data Collection and Presentation
> Control Applications.

More detailed consideration of appropriate activities which might take place within these categories is outlined in the paper. It is likely that the full range of IT strands identified in the National Curriculum may be accommodated within these design and technology subject based categories. Whilst such potential may be identified, however, the *application* of IT capability has not been achieved effectively. After some scrutiny into the implementation of Technology at Key Stages 1,2 and 3 HMI reported that:

Visits to all schools showed that:

> most IT work involved the manipulation of text. Pupils had little opportunity to use IT for modelling, control work or handling data.

And that

> Pupils often used IT successfully as part of their work in other subjects. However, schools need to develop all the strands of IT capability as outlined in the Order. [3]

Considerations based on categories that highlight the potential for IT to support a curriculum subject should help teachers to focus more clearly on the IT skills to be developed, and also to identify how they will be applied. Design and technology provides a wide range of opportunities for these skills to be developed and applied within the realistic context of designing and making activities, and in a manner commended by non–statutory guidance.

With such a clear potential for IT to enhance designing and making activities, whilst at the same time addressing the observations of HMI, it is not surprising that many design and technology teachers are considering the adoption of suitable hardware and software. Unfortunately, available software does not always meet the needs of teachers and pupils. This is particularly true in the area of design and technology work where it is normal for schools to develop their own resources rather than adopt published teaching schemes like 'Peak Mathematics' or the 'Nuffield Science Scheme'. More appropriate software would help teachers of design and technology to implement IT more effectively. It is particularly important that teachers are in some way involved with the development of educational software, and that they are more critical of software that fails to meet their needs. Greater consideration of the subject based IT needs, rather than the needs of IT across the curriculum, will help to determine the nature of IT support required and the type

of computer software that needs to be developed.

Consideration of the nature of design and technology, subjective evaluation of potential versus actual application of IT, and an awareness that only teachers can identify some of the elements required in educational software, have led me to outline some key criteria for design and technology software:

> The software should enable learning within an identified context.
>
> The extent of software 'flexibility' should be appropriate to the intended application.
>
> The software documentation and support should be aimed specifically at pupils and teachers in schools.
>
> It should be possible to operate the software effectively after a minimum of teacher instruction.
>
> The quality of output material should be consistent with the required attributes of design and technological activity.

The evaluation of educational software — teacher participation

Any analysis of how pupils learn, and the evaluation of resources that are designed to enhance their learning, is a complex business. It is, however, worth concentrating on three of the issues involved with the process of evaluation since they are central to the development of effective educational software and highlight some questions concerned with the teacher's role in software development and evaluation. Teachers inevitably find it difficult to allocate substantial time to assist with software development.

Is it reasonable or realistic to expect teachers to be substantially involved in the development of educational software?

The information gathered from teacher evaluation, particularly the formative evaluations described, will have been coloured by the very act of evaluation itself. It is difficult to measure the performance of educational software without influencing the normal context of its use:

> The major threat to the validity of both quantitive and qualitative research approaches lies in the concept of reactivity. This refers to the tendency of any research process to distort the reality that it seeks to investigate... it is a rare participant–observer who can enter and leave the research field without creating any disturbances. [4]

Is it possible to minimise these 'interventionist' effects of software evaluation?

Early discussions between potential users and software developer have been particularly beneficial in my experience of designing CAD/CAM software. Development of an early prototype, albeit with limited functions of operations, was at the centre of these discussions. It was able to highlight the nature of interaction between pupil and software, the features that should be included and how these features should be provided so that particular needs of pupils in school might be accommodated. Only teachers and pupils can provide this insight, yet a dialogue with software developers is often neglected at the early stages of design.

The small market for potential sales of educational software makes it difficult to justify the cost of protracted consultation, but development of an early prototype can provide a catalyst for discussion which can minimise this process.

Is it possible to provide prototypes during the initial stages of software development, and can they be provided in a cost effective manner?

The answers to these questions will help to identify the teacher's role during software development. It is a role that is vital to the successful development of design and technology software.

The teacher's role in software development — strategies for their involvement

The best use of teachers' time is to ensure that they concentrate only on the issues that the commercial developer may neglect. It is possible to separate the more technical appraisals from those of an educational nature. The software developer is able to check that a program runs reliably, provides accurate information and performs to an original specification. It is acceptable for these technical, more objective, appraisals to be carried out by people who do not have a strong background in educational practice.

Appraisals that are concerned with the quality of learning and a pupil's interaction with the computer are likely to be more qualitatively

based. Such evaluations need not be difficult if the evaluator has the background and experience to make meaningful judgements based on their general appraisal of a lesson. Teachers are able to do this, and their assessments are likely to be as accurate through the use of informal comments as through the use of 'objective' check lists.

Elements of software appraisal may be made from video recordings of its use in a lesson, allowing the teacher to concentrate on teaching the lesson and to make judgements about the software at a later time. This will make it more convenient to carry out the appraisal, but the video camera, like an observer, can be intrusive. Once again the software is not observed in its normal context. An alternative approach has been explored during final evaluation of CAD/CAM software development. Computer output was connected to a standard VHS video recorder in addition to the conventional monitor. A microphone was added to the system and recordings made during a normal teaching situation — it was not necessary to make special arrangements to accommodate the evaluation process. A complete log of pupils interaction with the software, together with their spoken comments, are recorded on video tape and used to assess the more technical (objective) elements of evaluation.

The development of circular arc drawing utility provides a useful illustration of this approach. Some software packages provide many different ways of drawing arcs, so that they suit a wide range of different drawing situations. Unfortunately, this can present pupils with a bewildering array of decisions to be made and data that need to be determined. The aim of my software was to provide an appropriate arc drawing routine for CAD/CAM tasks in school, and analysis of video sections showing arcs being drawn by pupils identified the following points:

pupil generally had a sketch/drawing of the shape they wished to draw on screen;

the sketch would indicate each end point of an arc, but not usually the centre;

precise measurements of arc radius were sometimes identified but most pupils preferred to judge the curve's radius 'by eye'.

Final versions of the software implemented the following arc drawing procedure:

1. Select clockwise or counter-clockwise arc using an icon.
2. Fix the start of the arc with a mouse button click.
3. Fix the end of the arc (providing a default arc in the first instance).
4. Adjust the radius by 'banding' or enter the radius via the keyboard.

Other elements were evaluated from the teacher's subjective appraisal of educational issues. For example, the level of teacher support that was required before pupils could successfully design and manufacture products; the nature of that support and any other resource required; the quality of design outcome.

This was found to be a most successful evaluation strategy, gathering a wealth of relevant information whilst minimising the teacher's role and intervention in the lesson. This approach starts to answer the first two questions raised earlier.

The third question is concerned with the need to enable meaningful discussions between teacher and developer at an early stage of the software design. The CAD/CAM software was able to use a modified version of a similar application to serve as a catalyst, but this approach could not be generally adopted. Software tools do exist, however, in the form of 'authoring applications' which might serve a similar purpose. The authoring application allows images and text to be specified, manipulated and presented with great accuracy and ease. Teachers who are used to 'Hypercard', 'Genesis', 'Magpie' or 'Top Class' will be familiar with the approach and style of an authoring application. They may even teach their pupils to present information on similar systems. As the systems become more sophisticated and even easier to use, they provide an ideal means of simulating the style, presentation and anticipated learner interaction with the potential software. Screen layouts, software response and the learning options required may be considered by teacher and developer, both of whom are able to make modifications to the prototype as discussions proceed. Teachers might even specify new product requirements through the use of authoring applications, leaving the more tedious program coding for the commercial software developer.

References

1. DES (1990) *Technology in the National Curriculum.* p.C4 IT 2.3, HMSO
2. Hodgson, A.R. (1992) in Smith, J.S. (ed.) *International Conference on Design and Technology Educational Research and Curriculum Development.* pp.131-135, Loughborough University,
3. DES (1992) *Technology 5 to 16.* p.9 and p.36, HMSO
4. McCormick, R. and James, M. (1989) *Curriculum Evaluation in Schools.* p.191, Routledge