Developing computer software applications for use in Design and Technology Education

A R Hodgson
Department of Design and Technology, Loughborough University of Technology

Abstract

There is little doubt that design work provides a natural focus for Information Technology (IT) activities and that both profile components of National Curriculum Technology are intended to be centred upon real applications and real situations.

Despite a plethora of computer software applications which might be considered relevant in the context of Design and Technology work, few are rarely designed to meet the needs of pupil learners and so it is not surprising that few are used effectively. If IT is to become an effective tool in education, there is an urgent need to consider how pupils and students learn through interaction with computer media, and ways in which the teacher's role might be developed. These issues are quite different from, but no less important than, those considered by the software engineer, who is predominantly concerned with providing elegant program code and sophisticated program facilities.

This paper discusses how these two sets of, often contradictory, matters may be corporately considered to provide more effective software design. In particular, it considers how research, development and evaluation, concerned with computer applications, might take more effective account of educationalists' views, teachers' requirements and pupils' needs.

Introduction

This paper is concerned with the quality of educational software which is available to teachers of Design and Technology in schools. It considers the ways in which teachers and learners may take advantage of this software and the need to merge the views of teachers and software writers in order to make more effective use of computing resources.

Software in schools has always been open to criticism. Educational software has only a relatively small market and must be provided on a small budget and at low cost to the end user. Until recently, it was characterised by "cut-down" existing commercial packages, which aim to emulate computer use in the "work place", or by amateur programs created by well-meaning teachers who have become frustrated by a lack of educationally appropriate software. The paper written by Self [1] five years ago is typical of the criticism which software has received. His paper, "The Institutionalisation of Mediocrity and the Influence of Outsiders", was particularly pessimistic about the future of educational software. It began by stating that

> . . . most educational software is of poor quality . . . and many others have come to the same conclusion . . . although the castle-building 'teacher programmer' will, in fact, detect many signs that his products are of high quality.

Many would agree with Self, suggesting that teachers have neither the time nor experience to write effective software and that commercial software development often fails to take account of educational needs. If we are to move away from these criticisms and towards the development of computer software which makes effective use of IT in Design and Technology, then we should consider some of the issues which must guide its development and determine how they may be incorporated in that development.

Educational context

Software which is developed specifically for an educational purpose should take full account of the educational context in which it will be used. Inevitably, this will include consideration of the curriculum and its subjects, the potential for computer applications to aid learning and the role of teacher and learner when using IT resources. We might expect the school curriculum area described as "Design and Technology" to be adequately covered by the statutory orders for Technology in the National Curriculum. They are, however, the subject of some controversy and much mis-understanding. In fact, I believe that many more conferences and much re-writing will be required before National Curriculum documents describe fully, or accurately, the nature of design and technological activity in schools. Fortunately, interim reports of the working group which developed the orders provide a useful description.

> . . . view technology as that area of the curriculum

> in which pupils make useful objects or systems, thus developing their ability to solve practical problems . . . They should be taught the principles and practice of good design, the application of theoretical knowledge and, within that context, the practical skills for realising their designs in wood, metal, plastics, textiles and other materials [2]

Technology in the National Curriculum has moved pupil activity away from the acquisition of making skills for their own sake towards the development of design-based activities in which these skills are developed in order to "realise" a design. In this approach the acquisition of "making skills" should be viewed quite differently from that which may be associated with more traditional or vocational needs. Pupils may require skills which are specific to their individual needs and which may be irrelevant to the rest of their group - all of whom are also following individual design work. Design and Technology teachers need to consider alternatives to the "class-teaching" styles which were characterised by the whole group receiving instructions, describing how to make specific objects, and yet must still be concerned overtly with the development of skills and concepts.

Significantly, little specific content or few specific manufacturing skills are identified within the Technology document. It chooses, instead, to describe a design process and general capability.

The Design and Technology profile component challenges teachers to consider the nature of design and technological activity in order to determine what content, skills, knowledge and teaching styles might be appropriate in its delivery. It is important that learning activities are based on real situations (in the pupils' eyes!) so that pupils may see that there is a real need for decisions and a real purpose for their work.

The IT profile component clearly applies to all teachers across all curriculum areas and subjects, but shares a similar ethos and rationale to the four Design and Technology attainment targets. Information Technology should not be taught simply for its own sake, but through the need to apply it to real situations, which could be identified by pupils working in any area of the curriculum. This approach is supported by HMI who describe some aims of IT in schools thus

> It has a critical role in enhancing the learning process at all levels and across a broad range of activities . . . . Through the use of IT in the curriculum, schools will be helping pupils become knowledgeable about the nature of information, comfortable with the new technology and able to exploit its potential. . . . Using the technology to support collaborative working, independent study and re-working of initial ideas as well as to enable pupils to work at a more demanding level by obviating some routine tasks. [3]

The two profile components, and the five attainment targets of Technology, describe an area of teaching and learning which is characterised by a "holistic approach" to studies, through "context-led" activities which aim to focus upon the "progression of individual pupil capability". For many schools and teachers this is a new approach to learning, and the role which they adopt is crucial to the successful teaching of Design and Technology. It does not mean that all work should be entirely open, entirely pupil directed or content free. It will require teachers to consider, amongst other things, the relationship between "designing" and "making" activities; the balance between structured or focused and open project work; the extent to which pupil autonomy might be fostered; and how real pupil progression might be developed. These issues are at the heart of Design and Technology implementation but, perhaps more importantly, help us to focus upon the ways in which Information Technology might help to enhance designing and making activities in schools. The thoughtful application of IT to Design and Technology should help to overcome some of the difficulties encountered by teachers now charged with the task of National Curriculum delivery, and a clear understanding of the nature of Design and Technology in schools must be conveyed to those who develop software which is aimed to support pupil learning.

The potential for IT in Design and Technology

A clearer identification with established classroom practice may help the software writer to identify desirable features for inclusion or to prevent the repetition of past bad practice. The potential for IT in Design and Technology can be categorised through typical use rather than learning style. A range of such categories might be :

> Graphic communication
> Computer Aided Design and Manufacture
> Research / Data Collection and Presentation
> Control applications.

These are an appropriate starting point because they represent some of the ways in which computers have been used to aid designing and making in recent years. There is a need to identify the ways in which they may more positively aid designing and

making in schools, and to provide some guide lines for software developers.

Graphics applications are one of the most obvious, and certainly most common applications, in Design and Technology. Many 2D art and line drawing packages allow detailed colour images to be developed via the keyboard and/or mouse, and the advent of more powerful 16 and 32 bit computers has improved both the quality and realism of these images. However, the use of these applications should go well beyond the coloured "folio cover" or alternative "presentation drawing", so often a feature of their application in school. Their restriction to these types of application raises two particular questions :

Do computer graphics packages really make it easier or quicker to draw images than more traditional techniques?

What are the distinctive advantages of computer graphics packages over the other techniques which are used to develop and present ideas?

The first question is easily answered by any teacher who moves away from testing the attractive facilities of "graded tones" and "edge detection", to try and draw an image of a relatively complicated, realistic object instead. It is more difficult to draw freehand images with a mouse than with a pencil, and it takes longer to draw detailed component drawings by computer than by hand - particularly when the software operation has to be learnt as well. Most of us have come to realise that whilst it might, at first sight, appear to be attractive use of the computer, it is not realistic to draw images in this way simply to be printed out once and used to enhance folio presentation. It is only when these images are manipulated by the computer, or applied in a variety of ways, that the potential of graphics applications may be exploited. Good examples of their use exploit the software's ability to flip, rotate and mirror images; to alter their colour easily; or to make amendments to design detail without having to re-draw whole views.

Similar examples of the potential for computer applications may be applied to my other categories of potential use, and it is clear that teachers (not software developers) must take responsibility for ensuring the effective application of computer software in an appropriate learning environment.

Criteria for Software Development

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 most important criterion, in my view, is that a system should enable learning within an identified context. Although many other criteria may help to measure the system's suitability, there is a need for the teacher or student to identify the role, need or purpose of the software and then to relate these to the identified options or features which the software may provide. For example, it may be more appropriate to use a word processor designed for primary school pupils when introducing basic concepts of text manipulation to older students, rather than to make use of a sophisticated "office" system which has been designed to mail - merge, multi - task and communicate with spreadsheet systems. Identifying the learning context, and specifying appropriate software requirements, is an important first step which helps to clarify the relevance and priority of any further criteria used for development and evaluation.

The flexibility of software is a feature which is often considered with little reference to my first criterion, particularly by writers of software. A large element of flexibility, characterised by provision of many complex options, may not be appropriate and might even detract from the learning process. For example, students may wish to generate stylised text for use on a product package by using a computer aided Art or DTP system. Many of the features which are normally associated with these systems are irrelevant to their needs, and could confuse or divert from the task in hand. It may be useful in these situations, to provide an option for the teacher to "de-select" some features or aim software at specific tasks in a more overt manner.

The software documentation and support is of vital importance for educational users of software. It must address a range of quite different needs and the inevitable compromise is often disappointing. Although it is difficult to generalise, since the learning context might determine some very specific needs, I would identify three types of support as a minimum requirement :-

An overview of the software's features, options and aims. This may be provided through exemplar material and/or an outline of educational rationale.

Detailed or technical information which will enable the more experienced user to take full advantage of the software or, in the case of a teacher, to "tailor" the software to more specific needs.

Simple guides, reference or tutorials which enable "naive users" to make effective use of the software. This would include information related to the loading and running of the software. These elements of support might more appropriately be produced by teachers rather than software developers.

Although the criterion that it should be possible to use software effectively, after a minimum of teacher instruction, might relate to many areas of the curriculum, it is one which I believe is particularly important for the subject area of Design and Technology. The National Curriculum has identified the role of the teacher a an "enabler / advisor" rather than "trainer / instructor", with pupils taking greater control of their own learning. One outcome will be a wider variety of learning routes and needs within a single teaching group and the opportunity for teachers to explain how software operates to a "whole class" may disappear. Information generated by the software itself (possibly in the form of screen displays / instructions) together with relevant documentation and additional resources may be the only "instruction" which a learner receives.

Another criterion which may be specific to Design and Technology is that the quality of output material should be consistent with the required attributes of design and technology activity. This subject area will often use software to produce "end products" directly. For example, advertising projects may use DTP or CAD output directly on packages, products or posters, and CNC knitting machines may be used to design and make a garment. Consequently, software of this type must be capable of producing output which is accurate and precise enough to be used in a realistic end product or artefact.

These five criteria may usefully form the basis of software development and evaluation in Design and Technology.

The way forward

Schools already possess computer hardware which will serve as a platform for powerful and sophisticated software applications. At a time when increasing emphasis is being placed on "Technology and Industry", we must not allow these new platforms simply to run the software of industry and commerce. There is a need for software development to take account of the criteria outlined above, and in particular, to take greater account of teachers' views and experiences.

As the hardware becomes more powerful, the teacher cannot, and should not, expect to become accomplished "programmers" themselves. Their role of advising, enabling and evaluating is vital, however, and we must find ways and means of combining their educational experience with that of the software engineer / developer. Whilst this already happens in a few instances of good practice, the quality and range of software available for Design and Technology teachers is generally poor, and I would offer the following suggestions as possible ways forward :

Teachers should be involved, significantly, in the development and evaluation of software - particularly the formative evaluations which lead to the final version. This need not detract from normal classroom teaching or involve substantial teacher time. For example, pupil interaction with a software application may be recorded on video, allowing teachers to work normally in the studio / workshop and evaluate software features at another time.

Teacher training should highlight how fourth generation programming tools may be used to ease the development of software resources, or to modify existing software so that it becomes more suitable for a specific learning situation. For example, "HyperCard" or "Genesis" systems are easily accessible to the computer - literate teacher, and do not require the knowledge of expert programmers to be used effectively.

Software should allow more opportunity for teachers to "tailor" or "configure" its features / operation to meet specific educational needs. For example, more complex features may be "hidden" from the user, complex notation written in more familiar

terms and sensible "default settings" all determined by the teacher through a simple software option or facility.

Good examples of computer software already feature significant teacher involvement in their development and teachers of Design and Technology are ideally suited to adopt a more positive role in software development. They already possess background in design and development, are usually quite accomplished software users and are used to developing much of their own teaching resources.

References

1 Self, J *in:* Scanlon, E and O'Shea, T (Ed.) *Educational computing.* Wiley (1987) p 221

2 DES *National Curriculum : Design and Technology Working Group. Interim report.* HMSO (1989) pp 86-87

3 DES *Information Technology from 5 to 16 : Curriculum matters 15.* HMSO (1989) p 2

Bibliography

DES *Technology in the National Curriculum.* HMSO (1990)

DES *Craft Design and Technology from 5 to 16 : Curriculum matters 9.* HMSO (1987)

Barber, J and Tucker, R *The Interactive learning revolution.* Kogan Page (1990)

Fothergill, R *Implications of new technology for the school curriculum.* Kogan Page (1988)