Reading (1986) **20** (2)
0034-0472 $2.00

Too Much Software? An update on computer-assisted learning in language and reading.

David Wray

Background

Early writing on the use of the microcomputer as a means of developing reading and language tended to concentrate on outlining possibilities and suggesting activities which the computer might be used to support (Wray, 1983; Obrist, 1983). In the first flush of enthusiasm, much was written about the potential of the computer in educative settings, but attempts to evaluate work actually being done in classrooms came much later. This was, to an extent, inevitable for two reasons. Firstly, as with any exciting new development in education, or indeed elsewhere, heady optimism is a natural response of those committed to the idea of the innovation, and vision rather than caution is the order of the day. Secondly, and more practically, the early commentators were writing within a context in which ideas were far more plentiful than concrete examples. There were simply not enough schools with access to computers and not enough software to use with these computers to be able to evaluate ideas on any but the smallest of scales. Case-studies were published (Jones, 1982) but by their very nature these tended to involve highly committed participants: in other words, those who had been sufficiently interested from the beginning to acquire the hard and software, when the vast majority of the educational community had barely begun to consider computers as an issue.

CONTACT THE AUTHOR':
Lecturer in Primary Education, University College, P.O. Box 78, Cardiff, CF1 1XL

The situation has now, however, changed completely. Evaluation has begun to assume its proper place in the development process. Several projects are underway, or have reported, which attempt to evaluate aspects of the use of computers (Broderick & Trushell, 1985; Potter, 1985). Case-studies are still published (Potter & Wray, 1985; Open University, 1984), but we now have available a much larger range of examples of computer usage, in terms of subject, age of children and, very significantly, expertise of teacher.

The hardware and software problem has now also been resolved. Because of Government sponsorship every school now has access to at least one microcomputer. Primary schools form the largest group of computer users, and publishers have not been slow to exploit this new market for educational software. There has been an enormous growth in the quantity of software available, much of it intended to assist the development of reading and language. Unfortunately much of this software is inappropriate and just plain bad. Teachers may be forgiven for being overwhelmed by the sheer quantity, with even software from reputable sources such as the M.E.P. occasionally being questionable in its content. From a situation in which there was too little software available, we have rapidly reached the unthinkable: there is actually too *much* software available, so much so that it is an impossible task to keep up with what is there. Yet when consideration is given to what software teachers and children actually need, the list can be narrowed considerably. The rest of this article will argue that, as far as the development of reading and language is concerned, the teacher actually only needs *five* types of software, from which endless possibilities for worthwhile activities will spring. Before outlining what these five types of software are, some time will be spent, by way of general justification, in outlining some criteria by which this list was arrived at. These criteria may also be found useful for the evaluation of new software as it becomes available.

Criteria for software evaluation

By taking into account the ideas of several commentators, and also by applying what is generally accepted wisdom about other areas of the language and reading curriculum it is possible to arrive at five criteria by which good software for developing reading and language can be judged.

1. Openness. Good software, like most good educational materials, should be open-ended. This can be interpreted in either of two ways. The software can either be content-free, like a word-processor, and simply operate as a tool (Anderson, 1984) upon content which is chosen by children or teacher, or it can be open-ended in the directions in which it might lead. An adventure game would meet this criterion, even though its content will be pre-determined, in that it

could be used as a stimulus for a wide variety of activities, according to pupil interests.

2. Stimulates creativity and problem-solving. Good software should cause children to *think*, rather than call for a series of conditioned responses in what Chandler (1984) has called the 'hospital model' of learning. The software may pose problems, like an adventure game or a simulation, or it may be a tool for solving them, like a data-handling program.
3. Can be used across the curriculum. Language development does not just occur in English lessons, and good software should encourage this across-the-curriculum language work. Software which operates as a tool rather than a delimiter of content may, again, be more effective at this. A word-processor can, naturally, be used to write about anything, and a good adventure game can stimulate work in areas ranging from science to creative arts. (Wray, 1985)
4. Flexibility. Software, like any teaching material, is only as good as the teacher who uses it. Even the best software can be used unimaginatively. But it is very difficult for imaginative teachers to use poor software very effectively. The point is that good software should be capable of being *used* by teachers, rather than using them. It needs to be flexible enough to be used in this way. Anderson's (1984) list of suggestions for the uses of the 'computer as tool' testifies that there is no lack of possibilities for the use of flexible software.
5. Encourages co-operation. Good software should provide a context in which co-operation and discussion, which we know to be helpful to good language development, can flourish. It has been demonstrated that there are arrangements the teacher can make to maximise this discussion (Potter & Walker, 1985), but no amount of organisational manipulation will make up for the poor stimulus for discussion provided by some commercially available software.

Essential pieces of software

There seem to be at least five types of software which meet the criteria above and would seem essential for every primary classroom to have available.

1. *Adventure games / Simulations.*

Most teachers will recognise the power of the adventure game and of the simulation in stimulating problem-solving in children, and also in encouraging great amounts of follow-up activities in a variety of curriculum areas. There are several case-studies now available of classroom work with adventure games involving fantasy (Potter and Wray, 1985) and with simulations based upon realistic situations (Whittington, 1984). Of particular note is the emphasis which has been placed upon the stimulus afforded by this type of program to co-operative discussion (Bleach, 1985). Less often considered, but of equal importance, is the great emphasis these programs put upon the ability to read very carefully, and often to make inferences on the basis of that reading (Wray, 1985). Children's motivation to read the screen can be immense. The problems which might be posed by this screen—reading have not, as yet, received adequate study. It has also been suggested (Hart, 1984) that there is a great deal of potential in encouraging children to devise and program their own adventure games. A flexible, user-friendly adventure-creating program would seem a high priority in terms of essential software.

2. *Word-processing.*

Current thinking on children's writing stresses the encouragement of reworking and re-drafting (Graves, 1983). The problem with this is, of course, the sheer physical drudgery of writing and re-writing. A word-processor takes away the physical problems of writing and can genuinely free children to compose and enhance the quality of their writing. Studies are beginning to be undertaken into the uses of word-processors in schools (Broderick & Trushell, 1985), and, if the reported enthusiasm of teachers and children for their use is repeated elsewhere, the word-processor would seem to have a very definite future in a range of educational contexts. Its success has been reported with very young children (High & Fox, 1984), as well as with juniors, and there is work currently in progress in American schools (Wallace, 1985) which suggests that, given the simplification of writing with a word-processor, young children can use this approach as a way into initial reading. Keane (1984) also shows that the potential of a word-processor is not limited to its use in composition, and suggests many ideas for its use in the language classroom.

3. *Data-handling.*

Handling information is already part of most children's experience in schools. Almost all teachers will at some point ask children to collect information and display it, an example being the project on ourselves which has as an end-product graphs of children's heights, weights, spans etc. Again the computer is ideally suited to tasks of this nature, and can allow children to handle data in quantities and with results that they could not otherwise manage. The adaptation of the 'ourselves' project to the computer has been described by Johnson (1984), and Ross (1984) has demonstrated that the computer can make possible the analysis by children of complex collections of data which would otherwise be beyond them. Data-handling programs and databases seem ideal for use as an adjunct and extension to project work in primary classrooms (Jackson, 1985). There are several pre-formed databases available, especially for use with the QUEST data-handling program, and these can be very useful for practising skills of data manipulation and analysis. Even more useful, though, may be the type of program, of which QUEST is an example, which allows children to design and construct their own databases in order to answer particular questions which they themselves have identified (Wray and Sharples, 1986). This is an example of what Anderson (1984) has referred to as 'computer as tool'.

4. *Information presentation.*

As well as gaining practice in the design, construction and manipulation of collections of information, children should also become familiar with computerised means of presenting information, that is, with the tools of information technology, such as Teletext and Viewdata. Use of these systems will gain in importance as time goes on, and their use is already an essential feature in many walks of life. Children's use of them may involve accessing information, and there are some very definite new skills which need to be acquired and practised in this regard. Their use should not, however, be limited to this essential, but fairly passive purpose. As with databases, some of the most benefit will be gained when children are encouraged to design and present their own 'pages' of information using these systems. Very little has yet been written about the use of this kind of program with children. Jackson (1985) describes the use of the EDFAX teletext emulation program as part of a piece of primary project work. The use of this program and others like it in the compilation of school magazines, project reports, and as a vehicle for information-exchange between schools is increasing, but does need much more study and an elaboration of its many possibilities.

5. *Planning logical procedures.*

It is becoming clear that, as well as being taught by, or learning through the computer, children should also be given the chance to teach the computer to do things in a logical manner. Anderson (1984) characterises this as 'computer as tutee', and this implies some form of programming being done by the children. Possibly the computer language most suited to this is LOGO, and a great deal has been written describing the possibilities of this language as a vehicle for problem-solving and for developing a healthy attitude towards mistakes. Papert (1981) suggests that a major benefit of the use of LOGO with children is that it can persuade them that making mistakes is a challenge, rather than something worthy of punishment. He suggests 'debugging' is an essential activity for children, and one which they generally will not need to be directed to do. It is, perhaps, unfortunate, that most of the accounts of the use of LOGO with children tend to put it into the context of problem-solving in Mathematics (Cooper, 1984). Some accounts do hint at its usefulness as a tool 'to promote language interaction among children' (Marshall, 1984), and it is this aspect which assures its inclusion in this list of five essential types of software for developing language and reading. Language development will always be a major aim of any co-operative problem-solving situations into which children are placed, and one of the real benefits of LOGO is the degree of self-setting of problems which seems to occur. If LOGO 'workcards' become popular, a great deal will be lost.

Conclusion

This article has attempted to do two things. It has firstly outlined a set of criteria which it is hoped may be found useful by teachers in selecting and evaluating educational software from the vast range which is now available, particularly that which purports to develop language and reading. Secondly, it has put forward the idea that there is, in fact, a limited range of software that is really essential for use with children. The fact that five program—types are described does not, of course, mean that only five *programs* will be needed. Each of these types may imply several programs, especially when the needs of different age-groups are considered. For example, with word-processing programs, a program which is adequate for infants (PROMPT for example) will not be for juniors, who may need to graduate to EDWORD or one of the other ROM-based programs. Similarly, a simple adventure game like MAGIC ADVENTURE may satisfy the needs of top infants, but top juniors will need something like SPACEX or FLOWERS OF CRYSTAL. What the list does is suggest a minimum requirement for most classrooms, and also, hopefully encourage a very critical approach to other types of software. There may well be gaps in the list (TRAY, for example, will not fit into any of the five categories, but has a great deal of potential for language development.) Teachers will, of course, use their discretion as to which other programs they wish to use with their children.

References

Anderson, J. (1984) The computer as tutor, tutee, tool in reading and language. *Reading* 18(2) pp. 67–78.

Bleach, P. (1985) Using Magic Adventure in the classroom. In Potter F. & Wray D. (Eds) (1985) *Micro-Explorations (1)* U.K.R.A.

Broderick, C. & Trushell, J. (1985) Word processing in the primary classroom. In Ewing J. (Ed.) *Reading and the New Technologies* Heinemann.

Chandler D. (1984) *Young Learners and the Microcomputer* Open University Press.

Cooper, M. (1984) Logo at Old Oak. In Open University *Micros in Action in the Classroom* Open University Press.

Ewing, J. (Ed.) (1985) *Reading and the New Technologies* Heinemann.

Graves, D. (1983) *Writing: Teachers and Children at Work* Heinemann.

Hart, R. (1984) The tombs of Arkenstone. In Open University (1984) *Micros in Action in the Classroom* Open University Press.

High, J. & Fox, C. (1984) Seven year olds discover microwriters. *English in English in Education* 18(2).

Jackson, M. (1985) Integrating the microcomputer into information—skills teaching. In Avann P. (1985) *Teaching Information Skills in the Primary School.* Edward Arnold.

Johnson, J. (1984) Using the computer for data-handling. In Open University *Micros in Action in the Classroom* Open University Press.

Jones, R. (Ed.) (1982) *Five of the Best—Computer Programs in Primary Schools* Council for Educational Technology.

Keane, P. (1984) Ways with word processors. *Times Educational Supplement* 9th. November, 1984, p. 53.

Marshall, P. (1984) Logo at Clitterhouse. In Open University *Micros in Action in the Classroom* Open University Press.

Obrist, A. (1983) *The Microcomputer and the Primary School* Hodder & Stoughton.

Open University (1984) *Micros in Action in the Classroom* Open University Press.

Papert, S. (1980) *Mindstorms: Children, Computers and Powerful Ideas* Harvester Press.

Potter, F. (1985) *Classroom organisation and group discussion: the role of the micro-computer, the role of the teacher* Report to Centre for Educational Research and Development, University of Lancaster.

Potter, F. & Walker, S. (1985) Organising group work with a micro. In Ewing, J. (ed)

Potter, F. & Wray, D. (Eds) (1985) *Micro-Explorations (1)* U.K.R.A.

Ross, A. (1984) The strongest conker in the world. In Open University *Micros in Action in the Classroom* Open University Press.

Wallace, J. (1985) Write first, then read. In Ewing J. (Ed.)

Whittington, I. (1984) Ponies, adventures and archaeological investigations. In Open University (1984) *Micros in Action in the Classroom* Open University Press.

Wray, D. (1983) Computer-assisted learning in language and reading. *Reading* 17(1), pp. 31–36.

Wray, D. (1985) The adventurous way to use the computer. In Ewing J. (Ed.)

Wray, D. and Potter, F. (Eds) (1986) *Micro-Explorations* (2) U.K.R.A.

Wray, D. and Sharples, J. (1986) Using a database in the primary school. In Wray D. & Potter F. (Eds) *Micro-Explorations* (2) U.K.R.A.