

Are we recognising the organisational impact on educational software design?

R. C. MacGregor
Department of Business Systems
University of Wollongong

Introduction

When computers were first used in business and industry in the 1950s, programmers and engineers were, for the most part, the only real computer users. Computer useability was little more than a question of speed of processing and throughput. As such, the idea of human computer interaction was a largely meaningless concept, as was any thought of the effect of the computer upon work practices. Grudin (1990) suggests that since that time the concept of human-computer interaction has shifted outwards from hardware, to software, to the terminal and on towards the work setting. Today, there is substantial interest in the effect of computer technology on organisations. Although the effect of computers on organisations is far from predictable, (Attewell and Rule 1984, Soergaard 1988) it is well established that the implementation of computer technology brings with it changes to the organisation and changes to relationships within the organisation (Little 1990).

While the outward movement suggested by Grudin has given rise to investigation of computer technology on users and organisations, it is only recently that interest has swung towards questioning the effect of the organisation on the design practices of computer systems. Hill (1988) considers that it is the economic, political, cultural and social conditions of an organisation which provides the context for design. Indeed, Little

(1990) suggests that 'many of the problems which arise during the course of technology projects stem from the lack of sensitivity to their organisational implications rather than any shortcomings in technical understanding' (Little 1990, 35.1).

Like any other technical endeavour, the design of educational software is grounded in the educational organisation. This paper examines the design practices employed in the production of educational software. It will be argued that while the educational organisation is the context of both software design and research, most designers fail to recognise its importance. It will be suggested that failure to consider the impact of the educational organisation on design practices can, and often does, restrict the learning process, rendering the final design more in line with the organisation than the learner.

An overview of education software

20th century education has provided a lengthy list of enthusiasms which have made quick entrances into education and, more often than not, even quicker exits. This phenomenon does not appear to be the case with educational software. The concept of using a computer or computing machine is by no means a new innovation. Indeed, if one considers the ideas of Skinner and Plessey, the use of such devices for learning actually predate the computer by several decades.

By the end of the 1970s categories of instructional software had become well established. However, if one is to consider educational software as a viable educative practice or, indeed, a discipline in its own right, concentration must be directed towards the past decade almost to the exclusion of previous effort. This is borne out in the literature (Bennet 1984, Eraut 1988) where it is suggested that categories of educational software established at the end of the 1970s appear now to be superfluous. As with any discipline the theories and practices adopted in educational software have had both their advocates and critics and it is appropriate to consider both in order to establish a realistic view of the current status of educational software.

The basis for the adoption of computer technology in education was the ability to offer individualised pacing and sequencing of material for the learner. The advent of the microcomputer and its affordability within the school system appeared to provide a vehicle for the individualisation of learning. Suddenly the problems of the one-to-many relationship (teacher to learner) could be replaced by a one-to-one relationship (learner to teaching surrogate). With advances both in technology and research, more and more course work has been directed to the computer alternative.

Today, educational software design 'taps into' areas such as touch sensitive devices (Tilley 1989), expert systems (Railsback 1986, Parr and

Wilson 1992), object oriented design (Henderson-Sellers 1992) and personalised interfaces (Salvendy 1987, MacGregor 1992). Along with advances in technology there has been a growth in the activity surrounding the utilisation of computing in the learning endeavour. These include work with the handicapped (Murche 1988), the exceptional learner (Hannaford 1986, Dover 1986) and specific cultures (Fleer 1989).

Given the increase both in technological advancement and the scope of learning potential, it is still apparent that many questions and criticism have remained unanswered. To fully understand the nature of educational software it is relevant to address these questions and criticisms and evaluate their effect and their causes.

There are many questions and criticism levelled at the design practices and research methodologies surrounding educational software. These questions include the validity of the parameters employed in research design (Moran 1981, MacGregor 1991), the approach to interface design (Eraut 1988, Tilley 1989) and the theoretical underpinnings of educational software research (Salvendy 1987, Hooley 1989). Indeed, some authors (Moran 1981, Mathias et al 1989) have suggested that much of the research is at odds with the real questions of learning.

Empirical studies of educational software design carried out by Riley (1984, cited Eraut 1988) have revealed a number of interesting trends. Riley compared actual design practices with literature writeups and found that many of the concerns confronting designers were not mentioned in the literature while many of the theories found in the literature were not evidenced in the software design effort. Riley further added that, all too often, designers and researchers were 'carried away with the sheer excitement of the technology' ignoring the more fundamental task of producing a viable learning environment. Eraut concluded that many assumptions and choices made by designers appeared to be out of habit rather than for any theoretically valid motive. Ultimately the questions and criticisms reduce to:

- the appropriateness/inappropriateness of the technology and software.
- the appropriateness/inappropriateness of the theory used in the development of educational software.
- the provision/lack of provision for the individual in educational software products and research.

To examine the impact of educational software design on the learner it is valid to adopt a simplified model of learning. The model chosen was originally used by Herbst (1974) to consider curriculum design alternatives. This model has been extended to incorporate the views of Riley and Eraut. Simply, the model suggests that if a learning task (T) is carried out by an individual, the individual's initial state (I) will be transformed to a new state (O) as follows:

T(I) -> (O)

This model is only useful if the task is fully determined, if we know precisely the initial state of the learner and if all the outcomes are fully measurable. If we adopt this model as the basis of educational software design then it could be argued that the model indicates the designers original premise. If, however, the software is designed in isolation from the learner and tested with insufficient parameters (Carrol and Thomas 1982) or little attention is paid to the learner as suggested by Riley and Eraut, then the model becomes:

T(?) -> (O)

This model suggests that we can accurately predict the outcome by simply knowing the tasks to be performed. In practice this style of software design gives rise to a product which does not 'live up' to promised outcomes. Thus our model becomes:

T(?) -> (?)

If, as suggested by Tilley (1989) and Mathias et al (1988) that the theoretical underpinnings of the software are questionable, the model further degrades to:

?(?) -> (?)

This final model is perhaps the one which best portrays the many unpredictable results found in the literature (see Williams 1983, Turner and Karasek 1984, MacGregor and Clarke 1989). It would seem, then, that the original designer's premise T(I) -> (O) is in fact a combination of:

T(?) -> (O)

T(?) -> (?)

?(?) -> (?)

While it may be argued that the design problems presented in the literature are, in part, attributable to an enthusiasm for the technology, the disparity between practice and writeups suggests that there is a need, at least in the writeup stage, for the designer to be seen to be producing an organisationally acceptable solution to the learning problem. If the organisation provides the context for technological endeavour, it must likewise influence research and design surrounding such endeavours. It would seem, then, that to adequately consider educational software we must consider the educational organisation in which it will operate. Furthermore, if, as has been suggested, design and research initiatives largely ignore the effect of the educational organisation, it is valid to re-examine criticisms found in the literature to determine if these criticisms are attributable to a neglect of the organisational parameter.

The nature of the educational organisation

Although educational organisations vary within and between levels of learning, a valid generalisation is that they are largely impervious to organisational change. This does not mean that there is no innovation, rather it implies that any change must fit into an already existing organisation boundary. As such, many changes of content and practice have been slow and often piecemeal such that the organisation can deal with them. It was into this slow changing environment that educational software, together with its individualised learning stamp, appeared. The result was a situation faced with two opposing perspectives: an organisation which lacked the capacity to change, and a recognised need for change such that learner directed practices were possible.

With an organisation unable to provide the necessary bases for educational software, designers and researchers were forced to look beyond education for the theoretical and practice foundations upon which educational software could be mounted. The gathering of theory and practice from other disciplines was often unwittingly premised on the immutability of the educational organisation. Thus rather than extend learning theory and practice, educational software became a subset to support rather than shake the existing organisational policy. Since these policies have been developed outside education and have been gathered to fit the existing organisational structure, their applicability to learning must be questioned. As will be seen, questions have begun to emerge as to the validity of appropriating theory and practice beyond the education confines, the effect of such practices on the learner, the content of the learning and on the design initiative itself. Indeed, it has been argued that since much of the theory, research and practice methods have been appropriated from other disciplines, without due recourse to organisational context, educational software design is not a discipline but a 'grab bag' of unconnected practices (Buter 1988, Keller 1987).

The effect on learning content of gathering theory and practice from outside the discipline

Ackoff (1974) suggests that the use of practices developed outside the education discipline only serves to disconnect the learning content. He adds that such an approach ultimately reduces the measure of learning to a measure of output and the time taken to achieve that output. The development of educational software and its subsequent research initiatives have, for the most part, maintained this approach. This is further exacerbated when one considers that most decisions concerning dissection, evaluation and design have not arisen within the discipline, but have been imported to support the existing educational organisation and are retained through historic precedence. Thus while learning has adopted

educational software as a tool, the theories upon which it is based are derived outside the field of education and in many cases are pre-computer. It is interesting to note that Moran (1981), in a discussion of computer aided learning describes it as having no function for internal evaluation and adds that it relies on 'folk psychology' for much of its theoretical content. Recent authors (Eraut 1988, Tilley 1989, Messing 1991) suggest that little has changed.

The effect on the learner of gathering theory and practice from outside the discipline

Most educational software is designed in isolation from the learner with an attempt to service as many learners as possible. This has led on the designer's total reliance on appropriated theories, norms, practices and evaluative techniques. In many cases stringent methodologies are supplanted by assumptions which reflect the designers own values resulting, at best, with a product tailored to a theoretical mean of the population. For the learner, there is not only a need to change learning habits to suit the software, but this change required of the learner is often used as a means of evaluation or categorisation. It is interesting to note that while the forcing of imposed standards in software design is criticised in the wider context (Turner and Karasek 1984, Damodaran et al 1988, Eraut 1988, MacGregor and Clarke 1989) because of its restriction to learning, these practices appear to be encouraged in the development of educational software.

It would seem, then, that when the learner is considered, questions must be raised concerning the erosion of individual differences and the replacement of these by theoretically defined and imposed standards. Furthermore, if, as suggested, these standards are appropriated from outside the discipline, their derivation appears to be coupled with a lack of recognition of the effect of the organisation on the design initiative. For the learner, removal of the asymmetrical dependence upon the teacher, the very foundation upon which computer use was based, comes with a price, resulting in less than the promised results.

In summary, design practices appear to contain the following weaknesses:

- Content is often dissected into unrelated fragments reducing learner control over goals and purposes of learning.
- Design is carried out in isolation from the learner and relies on arbitrary means of the population. This erodes individual differences and replaces them with imposed standards and imposed evaluation techniques, most of which are derived from outside the discipline.
- Little attention is paid to the impact of either the technology or the organisation on the design effort or the learner's perspective.

While much of the material presented thus far appears to be negative, it is comforting to note that recent literature has begun to provide alternatives. These will now be considered and, indeed, should form a minimum set of inclusions for educational software design.

- The impact of the educational organisation needs to be recognised and re-assessed by designers. This means that educational software design is not merely fitting pieces into the overall existing learning strategy, but rather re-assessing that learning strategy completely. Herbst (1974) suggests that this may entail re-examination and re-dissection of the total learning initiative such that new approaches are not fitted into an existing regimen, but rather the entire learning is based on the new technique.
- Design cannot be based upon a theoretical user. Indeed, Riley (1984) suggests that there is a very different agenda between the theoretical and the real learner.
- Design should involve the learner rather than being a task carried out in isolation from the learner (Eraut 1988).
- Designers must re-assess imported standards, many of which have little to contribute to research or design of educational software (Buter 1988).
- Design of educational software must be based on the context of the learning and must allow for individual methods of learning (Eraut 1988).
- Learning through educational software cannot simply be equated to time dependent output deliverables (MacGregor 1991).

Conclusion

Buter (1988) suggested that any evaluation of educational software design must conclude that there still remains a gap between theory, research, practicality and the organisation. While most of Buter's parameters have found exposure in the literature, the organisational parameter has been largely ignored. If technological innovation is embedded in organisational renewal (Turner and Karasek 1984) designers and researchers must begin to recognise the impact of the educational organisation on the techniques they are employing and the results they are attempting to derive. The designer must begin to question the theoretically derived standards of the learner population and place greater emphasis on the learner in situ. For the researcher, the exclusion of the organisation impact leaves the results 'up for target practice' since the research and its findings place learning secondary to organisational reinforcement. It is only by adding the organisational parameter that the true perspective of educational software design can be considered valid and it is only through recognition of the organisational impact that current criticisms become resolvable.

Bibliography

- Ackoff R. L. (1974). *Redesigning the future*. Wiley, NY.
- Attewell, P. and Rule, J. (1984). Computing and organisations: What we know and what we don't know. *Communications of ACM*, 27, 12.
- Bennett, N. et al (1984). *The Quality of Pupil Learning Experiences*. Lawrence Earlbaum Ass., NJ.
- Buter, E. M. (1988). Design: Scope and limits in the light of human sciences. In Mathias, H., Rushby, N. and Budgett, R., *Aspects of Educational Technology XXI: Designing New Systems and Technologies for Learning*. Kogan Page, London.
- Carrol, J. M. and Thomas, J. C. (1982). Metaphor and the cognitive representations of computing systems. *IEEE Transactions on Systems, Man and Cybernetics*, 12(2), 107-115.
- Damodaran, L., Ip, K. and Beck, M. (1988). Integrating human factors principles into structured design methodology: a case study in the UK. *Civil Service Eurinfo 88*, Athens, Greece, pp 235-241.
- Dover, A. (1986). Computers and the gifted: past, present, and future. In Harper D. O. and Stewart J. H. *Run: Computer Education*, 2nd ed. Brooks/Cole, California, pp 213-216.
- Eraut, M. (1988). What has happened to learning design. In Mathias, H., Rushby, N. and Budgett, R., *Aspects of Educational Technology XXI: Designing New Systems and Technologies for Learning*. Kogan Page, London.
- Fleer, M. (1989). Reflecting Australian Aboriginal culture appropriately in the design of educational software. School of Education, Canberra College of Advanced Education, in press.
- Grudin, J. (1990). The computer reaches out: The historical continuity of interface design. In Chew, J. C. and Whiteside, J. (eds), *Empowering People: CHI '90 Proceedings*, Seattle, pp 261-268.
- Hannaford, A. E. (1986). Microcomputers in special education: Some new opportunities, some old problems. In Harper, D. O. and Stewart, J. H., *Run: Computer Education*, 2nd ed. Brooks/Cole, California, pp 197-202.
- Henderson-Sellers, B. (1992). Object oriented analysis and design methodologies. In MacGregor, R. C., Clarke R. J., Little, S. E., Gould, E. L. and Ang, A. Y. (eds), *Proceedings of the Third Conference on Information Systems*. Department of Business Systems, University of Wollongong, pp367.
- Herbst, P. G. (1974). *Socio-Technical Design*. Tavistock, London.
- Hooley, N. (1989). Time to Assess the Value of PCs. *The Australian*, February 14.

- Keller, R. M. (1987). Concept learning in context. *Proceedings of the Fourth International Workshop on Machine Learning*, University of California, pp 91-102.
- Little, S. E. (1990). Quality and robustness in information systems: An organisational perspective. *Qualcon*, 35.1-35.15.
- MacGregor, R. C. (1991). *An information systems approach to the design of music compositional software for the young learner*. Unpublished Masters of Education (Hons) thesis, University of Wollongong.
- MacGregor, R. C. and Clarke R. J. (1989). User defined icon sets: A technique for the reduction of system complexity. In Karwowski, W., Genaidy, A. and Asfour, S. S. (eds), *Computer Aided Ergonomics*. Taylor and Francis, UK, pp 557-562.
- MacGregor, R. C. (1992). The development of music compositional software tailored to the young learner. *European CHI 92*, St Petersburg, pp 244-255.
- Mathias, H., Rushby, N. and Budgett, R. (1988). *Aspects of Educational Technology XXI: Designing New Systems and Technologies for Learning*. Kogan Page, London.
- Messing, J. (1991). The use of computer assisted learning in teaching programs. In Shipp, J. (ed), *Computer Assisted Learning in Libraries*. University of Wollongong, pp 19-28.
- Moran, T. P. (1981). An applied psychology of the user. *Computing Surveys*, 13(1), 1-11.
- Murche, J. (1988). Speaking up for the Disabled. *The Australian*, October 18.
- Parr, A. and Wilson, E. (1992). An expert system for the diagnosis of congenital heart defects in neonates. In MacGregor, R. C., Clarke, R. J., Little, S. E., Gould, E. L. and Ang, A. Y. (eds), *Proceedings of the Third Conference on Information Systems*. Department of Business Systems, University of Wollongong, pp 435-453.
- Railsback, C. E. (1986). Microcomputers: Solutions in search of problems. In Harper, D. O. and Stewart, J. H., *Run: Computer Education*. 2nd ed. Brooks/Cole, California, pp 113-114.
- Riley, J. (1984). The preparation of teaching in higher education: A study of the preparation of teaching materials at the Open University. D. Phil Thesis University of Sussex.
- Salvendy, G. (1987). *Human Factors*. Wiley, NY.
- Soorgaard, P. (1988). Programming Environments and System Development Environments. *Mjolner Report DK-SYS-31 Denmark DAIMI PB-252*. Department of Computer Science, Aarhus University.
- Tilley, L. (1989). *Controlling the genie: The human computer interface: Issues and opportunities*. A Discussion Paper. The Australian Science and Technology Council.

- Turner, J. A. and Karasek, R. A. jr (1984). Software ergonomics: The effects of computer application design parameters on operator task. *Performance and Health Ergonomics*, 27(6), 663-690.
- Williams, T. A. (1983). Technological innovation and futures of work organisation: A choice of social design principles. *Technological Forecasting and Social Change*, 24, 79-90.

Author: Robert MacGregor is a Lecturer in the Department of Business Systems, University of Wollongong.

Please cite as: MacGregor, R. C. (1993). Are we recognising the organisational impact on educational software design? *Australian Journal of Educational Technology*, 9(1), 59-68.
<http://www.ascilite.org.au/ajet/ajet9/macgregor.html>