

The Portability of Computer-Related Educational Resources: An Overview of Issues and Directions

Betty A. Collis and Italo De Diana

University of Twente, The Netherlands

Abstract

The portability of educational software is defined as the feasibility of software usage, with or without adaptation, in an educational environment different from that for which the software was originally designed and produced. Portability is a desirable characteristic of educational software from scientific, practical, and economic considerations. Many factors can affect portability; an overview of clusters of technical and human factors is given. The factors are further organized in a way that shows their interdependence. Research questions relating to educational software portability are given and illustrated with a seven-level analytical model. The chapter serves as an overview to this Special Issue in that the research questions and factors influencing portability that are defined within it are taken up, in various ways, by the authors of the following chapters. The concluding chapter of the Special Issue refers again to this opening chapter, as well as to a synthesis of the intervening work. (Keywords: software portability, technological change, computer utilization.)

Making educational software more transferable, or portable, is the focus of many research and development projects, in both academic and commercial settings, throughout the world. Educational decision makers from both more- and less-developed countries deal regularly with the issue of educational software portability: How can they best acquire the software they want? This Special Issue of the *Journal of Research in Computing in Education* focuses on the problem of educational software portability, from educational and international perspectives. The issue begins with this overview article, which sets the scene for the discussions which follow by presenting a general model of factors that influence the portability of educational software.

Portability. What is meant by the portability of computer-related educational resources? First, the concept of portability must be defined. This concept could be approached through a technical definition, such as: "The ability to

move an application from one computer to another unchanged and get the same results" (Dahlstrand, 1984, p. 17). However, the definition can also be expanded so that it can include both technical and nontechnical aspects of portability, for example: the capacity of an item to be used in an environment different from that in which it was developed. The purpose of this article, and one of the goals of this Special Issue, is to clarify and defend this broader definition of portability.

In many ways the term portability, as used in this global sense, overlaps other, perhaps more familiar terms and concepts relating to the reusability of educational resources—for example, transferability, technology transfer, and appropriate technologies. However, portability may be the term from this cluster whose initial meaning has been most closely associated with computer-related educational resources, and thus it, rather than one of the other terms, will be used in this Special Issue.

Computer-Related Educational Resources. The title of this article refers not only to portability but also to computer-related educational resources. Although the meaning of this latter term appears straightforward, it should also be defined and delimited. Computer-related educational resources could involve a continuum of possibilities, ranging, at one end, from human expertise related to computer use in education, to, at the other end, complex multimedia learning systems that include more than one computer and other hardware and software components. In addition, computer-related educational resources could also refer to print and multimedia support materials developed to support educational computing use. For this article, and most of the Special Issue, the focus will be directed at computer software for educational uses, although the issues involved have relevance to the entire spectrum of computer-related resources: electronic, print, and human.

The term *computer software for educational purposes* also needs delimiting. The category could, for example, include tools for professional software developers, as well as software products to be directly used by learners and teachers. For manageability, this article will again limit itself, this time to software products with which a learner or teacher is meant to interact. These products, for simplicity, will be referred to collectively as educational software. Thus software tools for professional developers will not be specifically discussed, although the portability of such tools is also an important issue. For example, the exploration of PETE, a portable educational tool environment, is a major aspect of the current European Community-sponsored DELTA Project, a large-scale research and design project involving both academic and industrial researchers from throughout the European Community and also the Nordic countries. (See FUNDESCO, 1990, for an overview of the first year of this project, and Nicklin, 1990, for a more specific discussion of preliminary user requirements for a PETE.)

Given these preliminary definitions, what are some important aspects of the problem of the portability of computer-related educational resources? To begin with, is there a problem?

ESTABLISHING THE PROBLEM: IS THERE A NEED FOR PORTABLE EDUCATIONAL SOFTWARE?

Portability is not a universal goal for all educational software relative to all new educational settings. Perhaps one of the most important wisdoms to develop with respect to assessing the portability of educational software is to know when it is advisable not to proceed with a particular portability project. However, there are important motivations for a general interest in increasing the portability of educational software.

Motivations related to cost, personnel, and time

One major motivation for an interest in educational software portability is more directly related to economics—the importance of saving cost and time—than it is to educational issues. Educational software development is a labor-intensive, costly process. Because of this, there is strong motivation to reuse components of software wherever possible. Wilson and McCrum (1984), for example, are among many who discuss the benefits relative to cost effectiveness and efficiency of using modular design and subroutine libraries in the production of educational software. Such strategies are, of course, well established among software engineers for both practical and theoretical reasons (Parnas, 1983).

Another motivation for efficiency relates to the need to improve the attractiveness of the educational software market. A steady improvement in the cost factors associated with educational software development is necessary in order to better stimulate the commercial market for educational software, both from the perspective of the producer and of the consumer (Oliveira, 1988; OTA, 1988). A recent survey in The Netherlands found, for example, that schools are only willing to pay a certain maximal amount for software (Ministry of Education, 1989, p. 23), even though the development costs of the software are considerably higher. Clearly, portability concerns, particularly relative to the reusability of various components of educational software and therefore to reductions in its costs, are an important consideration to both developers and users. Finally, portability relative to the potential sale of products to a broader audience than that for whom the product was originally targeted is also of obvious interest to software producers.

The economic arguments intensify in the context of small or less-developed countries. "Small countries simply cannot fund the major cost of their own educational software development" (IFIP Working Conference, 1989). Less developed countries are not only limited in financial means, but also experience a shortage of qualified manpower, scarcity of up-to-date development tools, and poor access to sources of technological advances (see Oualid, 1989, for an interesting discussion of these problems in the context of Arab countries). The shortage of adequate numbers of trained personnel to support a local software development industry can be a particularly strong motivation for software acquisition from outside. Also, "reinventing the wheel" not only is

costly and inefficient, but perhaps even impossible for small or less developed countries (Feinstein, 1989; Murray-Lasso, 1988). Many authors have discussed the importance of sharing educational software between more- and less-advantaged countries (Brauer, 1989; Hackbarth, 1985; Weston, 1989; Wolansky & Iyewarun, 1989) from both economic and moral (i.e., the belief that those who have should share with those who do not) perspectives.

Educational motivations

There also are strong educational reasons for the exchange and sharing of educational software. One motivation is simple—if a piece of software is good, then as many learners as possible should have the benefit of it.

Another motivation relates to the larger goal of contributing to the overall understanding of what makes educational software effective. Experiences with other educational media adapted for use in different cultures has yielded valuable insights into technology transfer. Friend, Galda, and Searle (1986), for example, were successful in translating interactive radio instruction originally developed for use in Nicaragua into a version appropriate for use in Thailand, because great care was taken in respecting the cultural and social differences in the two cultures and adapting the instructional materials accordingly. Comparing experiences with the same (or similar) software as it is used in various cultural and educational settings can make an important contribution to the relatively new field of educational computing. The benefits, and difficulties, in synthesizing experiences related to the impact of computers in education are substantial (Collis & Moonen, 1990); situations where the same or similar software is implemented and evaluated in different educational settings can bring valuable insights into the interaction of educational context with the characteristics of educational software and the effect of this interaction on learning.

Motivations related to the field of educational technology

From a scientific perspective, there are other motivations for the study of the portability of educational software. Careful analysis of problems that hinder the portability of educational resources and subsequent empirical research might lead to a clearer understanding about how to design for portability, and, more specifically, how to develop partial products, products that might be easily modifiable according to local circumstances, while still being mutually relatable. This type of knowledge could offer a valuable contribution to the growth of the educational technology knowledge base.

However, despite these educational and economic incentives, educational software portability remains a complex and unresolved issue, as the articles in this Special Issue demonstrate. Educational software portability is still a problem, and a problem whose solution is considered to be important. What then are some of the specific components of the problem?

SPECIFIC PROBLEMS ASSOCIATED WITH EDUCATIONAL SOFTWARE PORTABILITY

Specific problems involved with educational software portability can be loosely grouped into two categories: technical and human. Considerable investigation has already been done relative to these categories, particularly the technical category, although not too many projects involve the joint consideration of both perspectives.

Technical problems

Technical problems are perhaps easier to identify than human problems, as incompatible hardware and operating system environments are common problems confronting users of computers in educational settings throughout the world. Technical problems affecting software portability are the subject of much interest in both the commercial and theoretical world (Sandford, 1989) and a considerable amount of consensus is occurring over the need for standards with respect to software construction and architecture, hardware, and operating systems software. Technical problems affecting portability include: incompatibility of character sets for data; of data organization; of hardware-specific functionalities, particularly with respect to graphics; of languages and compilers; of memory requirements; of operating systems; of protocols for accessing peripherals.

There are still other technical factors that interact with contextual factors to limit educational software portability. For example, in less-developed countries an additional technical problem frequently arises because of the often lower memory and storage capacities of available equipment compared to the environments in more advantaged countries where software products were originally made. Murray-Lasso (1990), for example, described typical school computers in Mexico as 8-bit machines, the majority of which have memories between 16 and 64K, no disk drives or printer ports, and unreliable cassette memories. Software has not for many years been produced for hardware with these limitations in countries such as the United States, Canada, the Netherlands, or Sweden (all countries with commercial educational software producers who would be interested in the international sale of their software products). Planning for this sort of technical problem is an example of what Bengtsson (1989) called backward portability, in distinction to forward portability, which involves anticipating future advances in technical environments and conditions.

Among the subissues involved in the category of technical aspects of software portability, perhaps one of the most challenging relates to language—the language that the user sees, not the programming language used in the program. Translation of the text in the screen displays and user interfaces of a program from one language to another can involve different character sets as well as a conversion from textual to graphic format (Oualid, 1989; Zhuang & Thomas, 1987). It can also involve redesign of screen layouts because of the different

space requirements of different translations (IBM, 1987; St. Pierre, 1989; Trollip & Brown, 1987). And, of course, text translation is more than a technical problem, but also a cultural and social problem.

Technical problems affecting portability are also the objects of intense activity in the more general world of computing and software development, and standards in these technical areas will probably evolve, strongly influenced by marketplace pressures and industry-driven research and development rather than from models developed by educational computing professionals. However, given the current realities of technical incompatibilities, professionals in many fields have presented models or frameworks for software design that anticipate the management of technical problems within a design strategy. For example, a working group of the Interactive Video Industry Association has recommended that software be designed using a bottom-up, layered approach, in which only the bottom layer is defined in terms of hardware functionality. As another example, the Canadian province of Ontario has developed specifications for the so-called EASI (Educational Application Software Interface; Vraets, 1990). The EASI project is built upon a model of a seven-layer software portability environment, where the specifications of the EASI allow software developed for any of a variety of machines—but including references to the libraries, tools, and protocols of the EASI—to be translated by the EASI into executable code for any other of the six different hardware, peripheral, and operating system environments supported by the EASI (McLean, 1989; Ministry of Education, Ontario, 1987). Work on the Ontario project is now in its third year and still continues.

A final example of different approaches to the modelling of technical factors affecting software portability is derived from the virtual protocol model of computer-human interaction as presented by Nielsen (1986). Nielsen based his protocol on the reference model for Open Systems Interconnection (OSI) of the International Standards Organization for computer networks (1982). He defined seven levels of human-computer interaction—physical, alphabetic, lexical, syntactical, semantic, task, and goal. The first six of his layers relate to units of information exchange between the user and the computer, and he suggests design strategies to make the technical aspects of these layers within the computer system more flexible. His seventh level is external to the computer system, and relates to the real-world goals for which the software will be used. Stanchev, Nagtegaal, Collis and De Diana, and Murray-Lasso, in their chapters in this edition, make extensive reference to Nielsen's model.

Although work such as Nielsen's protocol and the EASI interface is making significant inroads into some aspects of the technical side of educational software portability, there is still a general overall limitation to software portability from a technical perspective. Fourteen years ago Bork made the following prediction about the importance of technical factors with respect to software portability: "When we consider the long-range point of view, no computer aspects of transferability are important" (Bork, 1976, p. 15). This prediction is encouraging and still seems warranted, but the long-range point of view when technical computer aspects become irrelevant, is nowhere in sight.

Human problems

The translation of the language of screen displays, mentioned earlier as a technical problem in educational software portability, is also an example of the second general category of considerations affecting educational software portability—considerations that can be called human problems. Language translation, for example, is more than the substitution of words and characters, but also involves context and meaning, inference and nuance. These are human, not technical, issues. In general, human problems affecting educational software portability include: educational issues relating to pedagogy and curriculum; organizational issues relating to the purchase, distribution, and maintenance of software; and social and cultural attitudes towards various aspects of educational computer use.

Educational issues are a well known category of human problems affecting educational software portability. Curriculum differences at either a micro (details relative to specific questions or educational objectives) or macro (the overall organization of scope and sequence of content in a given instructional area) level can make the content of a program inappropriate for another setting. Pedagogic or didactic differences also are important, including considerations such as the amount of time available for a teaching unit in different curriculums and timetables (Ballini & Polyi, 1988) and issues related to teachers' preferences for certain didactical approaches (Johnson, 1989).

Some issues related to teachers' preferences for certain didactical approaches are more personally or socially grounded than they are educational. Thomas (1987) noted, for example, with respect to teachers' acceptance of broadcast media, that a number of human reactions limit this acceptance. He discussed teachers' resistance to altering their traditional styles of instruction (either through inertia or through satisfaction with the style), teachers' dislike for outside interference, and teachers' fear of the apparent complexity of technology. Although such characteristics are not true of many teachers, they are true of some, and different cultural settings and traditions may magnify the characteristics when they do appear. Certainly the same problems Thomas cited with respect to broadcast media are pertinent to the transfer of computer-related media, within or outside of the country of its origin, and have the potential to form barriers to software usage as real as the barriers formed by technical problems.

Other portability barriers relate to social and cultural constraints. Murray-Lasso (the past president of the Mexican Society for Computers in Education) stated emphatically that "The first constraint is cultural. Mexico will not accept culture-dependent computerized educational materials that were developed for other cultures" (1988, p.3) because it cannot feel that it is being colonized or a satellite of somewhere else (1989). Wombi (1988) commented even more strongly: "Technology is like genetic material—it is encoded with the characteristics of the society which developed it, and it tries to reproduce that society" (p. 24).

Similar comments can be found in many sources. Ely (1989) noted that the way interactivity is handled in instructional materials must not be cosmetic, but must be done so that "cultural values are maintained and important traditions are not violated" (p. 12). Fler (1989) documented the considerable care and effort that went into the design of a software program for Australian Aboriginal children so as to better respect the culture and preferred cognitive modes of Aborigines. After describing a number of examples of cross-cultural projects where culture-related variables formed the major component of project success or failure, researchers from the Laboratory of Comparative Human Cognition (1986) concluded by saying "no simple translation between models applicable in one setting to models applicable in others can yet be made" (p. 1057). Ely made a similar conclusion when he said, "No formulas, guidelines, or published protocols will automatically lead to cultural and linguistic translations of materials from one country to another; it is ultimately the accurate and responsible communication of local value systems that determines the success of cross-cultural materials" (1989, p. 12).

And in addition to all these problems, there are still other human-factor barriers constraining the portability of educational software. Some others relate to the human organization surrounding software-related decision making and implementation. Who makes the decisions about software purchase and possible adaptation? How is dissemination about the software handled? Who develops the support materials, such as teachers' guides and student materials? How is software maintenance handled? How is pricing determined? How are copyright laws applied, especially if the software must be adapted to a variety of circumstances?

Finally, the cost of acquiring an appropriate product is also a factor that can seriously constrain the portability of any educational software. Regardless of the architecture of the software, the quality of the educational ideas involved, or the cooperation of the organizations involved in managing the software exchange, if the cost of the software relative to its expected benefits is too high for the buyer, portability will not be realized.

When all of these social, cultural, organizational, and educational issues are considered, Bork's prediction does seem supported—that the complexity of the human side of educational software portability will be more difficult to solve than that of the technical side.

CATEGORIZING THE FACTORS

The examination of technical and human problems related to educational software portability that has just occurred has made comment about a number of critical variables that can jeopardize the portability capacity of an educational software package. These factors can be listed more systematically if the two general categories of technical and human factors are expanded into the following list of technical, educational, social/cultural, and organizational factors (Collis & De Diana, 1990).

Table 1
Categorization of Factors Affecting Educational Software Portability

1. *Technical factors* relating to the:
 - 1.1 Physical input of data and mechanisms for human-computer interactions
 - 1.2 Program architecture, particularly as regards modularity and the separation of data from program structure, and algorithms, particularly with respect to language handling
 - 1.3 Authoring languages and tools used in developing the software
 - 1.4 Operating system level
 - 1.5 Hardware characteristics
 - 1.6 Network level (i.e., related to connecting the software with other objects, such as peripherals or other computers)
2. *Educational factors*:
 - 2.1 Educational need and relevance
 - 2.2 Curriculum content, globally and in detail
 - 2.3 Instructional approach, at the microlevel (i.e., How are individual questions handled?, What reinforcement is given?, etc.) and at the macrolevel (i.e., How is the class and lesson generally organized for instruction? What is the general approach of the software?)
 - 2.4 Tone and style of educational interactions
 - 2.5 Social and physical setting of the learning environment in which the software is planned to be used
 - 2.6 Teacher considerations, including their involvement in designing or adapting a program and the level of their experience
3. *Social/cultural factors*:
 - 3.1 Language in which the learner interacts with the software
 - 3.2 Tone and style of communication

- 3.3 Cultural identity
 - 3.4 Political sensitivities
 - 3.5 Cultural perception of the roles of the teacher and learner
 - 3.6 Local references and assumptions
 - 4. *Organizational factors:*
 - 4.1 Institutional procedures affecting software-related decision making and implementation
 - 4.2 Copyright and ownership
 - 4.3 Cost-related issues, including pricing, development costs, and cost effectiveness
 - 4.4 Marketing and distribution issues
 - 4.5 Maintenance
 - 4.6 Management of design, development, and distribution processes
-

OVERVIEW OF THIS SPECIAL ISSUE

Throughout the remainder of this Special Issue, attention will be drawn to these various factors. In addition, the summary chapter (De Diana & Collis) also uses the factors to summarize major trends and issues identified by the authors.

Implicit throughout all the articles in the Special Issue are the following research questions:

1. How can we better understand the critical factors that influence the portability of educational software?
2. How are the factors interrelated? How are they related in time and cost to the portability of educational software?
3. How can we deal with these factors to improve the portability of educational software?

In the concluding chapter (De Diana & Collis, 1990b) these questions will be considered relative to the contributions of the articles in the Special Issue. The productivity of the research questions as guides for further research will also be assessed.

Contributors

Betty Collis and Italo De Diana are members of the Division of Educational Instrumentation Technology at the University of Twente in The Netherlands. Dr. Collis works at the international level with the implementation and evaluation of computer use in education and training. Dr. De Diana specializes in the methodology of the design and authoring of educational software. Both are jointly involved in a 4-year international research project investigating the problems and possibilities of educational software portability. (Address: Department of Education, University of Twente, Postbus 217, 7500 AE Enschede, The Netherlands; FAX: 31-53-356531).

References

- Ballini, D., & Poly, A. (1988). *Working documents, The Summer University, Ecole Normale Supérieure de Fontenay-St. Cloud, 3-14 July 1988.*
- Bengtsson, B. (1989, June). *International perspectives on portability of educational software: The Scandinavian experience.* Presentation at the National Educational Computing Conference, Boston.
- Bork, A. (1976). *Transferability of computer-based learning materials.* Report with limited circulation, University of California, Irvine.
- Brauer, U. (1989). Educational software at the secondary level. *IFIP Newsletter*, 6(4), p. 9.
- Collis, B. A., & De Diana, I. (1990). International perspectives on the portability of educational software. In N. Estes, J. Heene, & D. Leclercq (Eds.), *Proceedings, The Seventh International Conference on Technology and Education* (pp. 581-583). Edinburgh: CEP Consultants.
- Collis, B. A., & Moonen, J. (1990). Assessing the impact of computer-related technology in education: Methodological strategies, issues, and new directions. In C. Bell (Ed.), *Innovations in assessing*. London: Kogan Page (in press).
- Dahlstrand, I. (1984). Software portability and standards. *Computers and their application*, 27, 1-147.
- De Diana, I., & Collis, B. (1990a, March). *Toward a model of educational software portability.* Unpublished report, University of Twente, Enschede, The Netherlands.
- DeDiana, I., & Collis, B. (1990b). The impact of different portability factors during the life cycles of an educational software adaptation project. *Journal of Research on Computing in Education*, 23(2), 306-317.
- Ely, D. P. (1989). Protocols and processes for promoting interactive cross-cultural media transfer. *Educational Media International*, 26(1), 6-12.
- Feinstein, B. (1989, June). *Software development and exchange: The Israeli experience.* Presentation at the National Educational Computing Conference, Boston.
- Fleer, M. (1989). Reflecting indigenous culture in educational software design. *Journal of Reading*, 32, 611-619.

- Friend, J., Galda, G., & Searle, A. (1986). From Nicaragua to Thailand: Adapting interactive radio instruction. *Development Communication Report, No. 52*. Arlington, VA: U.S. Office of Education, U.S. Agency for International Development.
- FUNDESCO. (1990). *Opening report: Trends and issues in DELTA*. Report submitted to the PRECISE Consortium, ECC, P.O. Box 217, 7500 AE Enschede, The Netherlands.
- Hackbarth, S. (1985). Instructional systems design: An appropriate technology for developing nations. *Programmed learning and educational technology*, 22(1), 35-38.
- IBM. (1987). *National language information and design guide. Volume 1: Designing enabled products, rules and guidelines*. White Plains, NY: IBM.
- IFIP Working Conference. (1989). *Notes*. Reykjavik, Iceland: IFIP Working Group WG 3.5.
- International Standards Organization (1982). *Draft International Standard, Information Processing Systems: Open Systems Interconnection—Basic reference model*. ISO/DIS 7498 (ISO/TC97).
- Johnson, N. (1989). CALL around the world. *C.A.L.L. Digest*, 5(1), 1-2.
- Laboratory of Comparative Human Cognition. (1986). Contributions of cross-cultural research to educational practice. *American Psychologist*, 41(10), 1049-1058.
- McLean, R. (Ed.). (1989). EASI software portability effort resumes. *CALM Development Newsletter*, 5(10), 1, 3-7.
- Ministry of Education, The Netherlands. (1989). *Use and valuing of the educational software coupon*. Zoetermeer, The Netherlands: Ministry of Education and Science.
- Ministry of Education, Ontario. (1987). *Functional requirements for micro-computers for educational use in Ontario schools: Stage II*. Toronto, Canada: Ministry of Education.
- Murray-Lasso, M. (1988, April). *Problems confronting the implementation of computers in Latin American educational systems: The case of Mexico*. Paper presented at the annual meeting of the American Educational Research Association, New Orleans.
- Murray-Lasso, M. (1989, June). *Portability of educational software: The needs and perspectives of developing countries*. Presentation at the National Educational Computing Conference, Boston.
- Murray-Lasso, M. (1990). Technology in education in Mexico. *ISTE Update*, 2(5), 1-2.
- Nicklin, P. (1990). *PETE (Portable Educational Tools Environment): The concept of PETE and some preliminary user requirements*. Paper prepared for the DELTA/EPOS Seminar, 22-23 January, Madrid. (Available from MARI Applied Technologies Ltd., Tower Street, Stockton-on-Tees, Cleveland TS18 3AQ, UK).
- Nielsen, J. (1986). A virtual protocol model for computer-human interaction. *International Journal of Man-Machine Studies*, 24, 301-312.

-
- Oliveira, J. B. A. (1988). Computer education policies in developing countries: Facing the hard choices. *Education and Computing*, 4 (3), 301-311.
- OTA. (1988). *Power on! New tools for teaching and learning*. Washington, DC: Congress of the United States, Office of Technology Assessment.
- Oualid, A. (1989). Problems of software development in developing countries. In G. X. Ritter (Ed.), *Information Processing 89* (pp. 1067-1071). Amsterdam: Elsevier Science Publishers B.V.
- Parnas, D. L. (1983). *Software engineering principles*. Report DCS-29-IR. Department of Computer Science, University of Victoria, Victoria, British Columbia, Canada.
- Sandford, N. (1989). Portability of interactive courseware. *Interactive Learning International*, 5, 125-130.
- St. Pierre, F. (1989). Presentation level issues are important when changing program to another language. *CALM Development Newsletter*, 5 (1), 6-7.
- Thomas, R. M. (1987) Educational radio and television—Their development in advanced industrial societies. In R. M. Thomas & V. N. Kobayashi (Eds.), *Educational technology: Its creation, development and cross-cultural transfer* (pp. 125-142). Oxford: Pergamon.
- Trollip, S. R., & Brown, G. (1987). Designing software for easy translation into other languages. *Journal of Computer-Based Instruction*, 14(3), 119-123.
- Vraets, J. W. (1990). Software portability: The Ontario approach. In N. Estes, J. Heene, & D. Leclercq (Eds.), *Proceedings, The Seventh International Conference on Technology and Education* (pp. 578-580). Edinburgh: CEP Consultants.
- Weston, C. B. (1989). Critical factors for educational technology interventions in developing countries. *Educational training and technology international*, 26(2), 122-128.
- Wilson, R. N., & McCrum, E. (1984). Use of modular design in the production of portable CAI software: A case study. *Computing in Education*, 8(2), 229-237.
- Wolansky, W. D., & Iyewarun, S. A. (1989). Effective transfer of technology in Third World countries: A systematic approach. *A.C.F.P. Journal*, 24(3), 9-12.
- Wombi, B. (1988). Domination by "Cooperation." *IDRC Reports*, 17(1), 24-25.
- Zhuang, J., & Thomas, R. M. (1987). Computer technology transfer to developing societies: A Chinese-language case. In R. M. Thomas & V. N. Kobayashi (Eds.), *Educational technology: Its creation, development and cross-cultural transfer* (pp. 65-90). Oxford: Pergamon Press.