The Portability of Computer-Related Educational Resources:

Summary and Directions for Further Research

Italo De Diana and Betty A. Collis

University of Twente, The Netherlands

Abstract

In this Special Issue of the Journal of Research on Computing in Education, the portability of computer-related educational resources has been examined by a number of researchers and practitioners, reflecting various backgrounds, cultures, and experiences. A first iteration of a general model of factors that influence the portability potential of educational software was given in the opening chapter by Collis and De Diana (1990b). In this final chapter of the special issue the general model is reconsidered, based on a synthesis of the intervening articles. Some factors were found to receive little notice by the authors of those articles; other factors were identified that were not explicit in the initial general model. Altogether, the perspectives of the authors complement each other and as a set support the validity of the model. (Keywords: software portability, cross culture, technological change.)

In the overview article of this special issue on software portability (Collis & De Diana, 1990b), factors that could be the components of a general model of educational software portability were discussed and a hypothesized list of the most salient factors was presented. These factors were organized in four groups: technical, educational, social and cultural, and organizational. How valid is this list of factors? How comprehensive? Should factors be added or reconsidered? How useful is the list approach as a tool to describe the system of factors that influence the portability of a software package? The purpose of this summary chapter is to address these questions through a synthesis of the issues and strategies raised in the 13 chapters in Sections II, III, and IV of this special issue and to suggest directions for further research in this area.

METHOD

The method used to synthesize the articles in this special issue was interpretive and analytical. The list of factors given in the overview chapter was used to form the 39 rows of a large 39 x 13 matrix where the columns of the matrix corresponded to the 13 articles between the overview chapter and this concluding chapter. Each guest editor, working independently, made notes and comments in the cells of the matrix indicating his or her impression of the attention given by the authors to each of the factors, the degree to which the authors appeared to feel the factor had an influence on the portability of an educational

software product, and the extent to which the authors felt the factor was manipulable by educational professionals involved with the software. The guest editors then compared their coding and comments, agreeing initially on 94% of the cells. Discussion of cells where there was disagreement identified situations in which the guest editors themselves disagreed on the interpretation of a factor, which happened three times.

Following their various iterations of discussions and analyses of the factor-x-article matrix, general conclusions were drawn. These conclusions will now be given in a summary form.

FACTORS AND ISSUES RELATED TO FACTORS

To test the validity of the selection of factors in the general model it is useful to note which factors were stressed by the authors, from both the points of view of the researcher and of the practitioner; which factors received little acknowledgement; which new factors emerged; and which factors were particularly identified as offering "solution areas" for portability improvement.

Factors Receiving Frequent Attention

Thirty-four of the 39 factors received emphasis from the authors, but eight of the factors appeared to attract the most frequent and extensive comments. These are: (a) factors associated with software architecture, particularly modularity, data/logic separation, and the technical handling of language-translation issues; (b) overall instructional approach; (c) teacher training; (d) cultural assumptions; (e) institutional decision making; and (f) the management of design processes. Modularity—the extent to which a program can be functionally decomposed so that any of its parts can be adapted without needing to alter its other components—was identified by many authors as an important technical factor. Eight of the 13 articles contained references to the importance of the choice of overall software type (i.e., tool-type, drill, simulation, etc.) as a critical component of later portability. It was frequently remarked that tool-type and content-free software are more likely to be portable than curriculum-specific software or software where instruction takes place through language-rich learner-program interactions. It was also noted by a number of authors that some curriculum areas were more cross-culturally similar—notably mathematics—and therefore better candidates for portability than more culturally referenced curriculum areas, such as history.

Teacher training, particularly so that teachers are better able to participate in the software adaptation process—during design sessions, field evaluations, or as end users—was also seen as important. Cultural assumptions, particularly those which differentiate between less-developed and more-developed countries, were also regularly stressed, particularly as the less developed country will frequently be the receiver rather than the originator of the product and thus may be more sensitive to the presence of assumptions in the software that reflect its source. Institutional decision making and management of the design

process were the factors most stressed from the organizational cluster, with several authors stressing the impact of managerial approaches.

Other “Hot Spots”

In addition to the above factors, which tended to be discussed in a broad and analytic fashion by the authors, several specific problem areas were frequently identified, often being described from the point of view of the teacher considering the use of software originating from somewhere else. It was frequently noted that this teacher may be frustrated if:

1. The program’s architecture is rigid and thus cannot be modified,
2. The product does not fit the local curriculum (and again, cannot be efficiently modified so that a better fit can be obtained),
3. A certain classroom environment appears to be presumed and this environment is different from that available to the teacher,
4. The teacher and students are not able to work in their mother tongue,
5. Certain cultural assumptions or orientations permeate the software that are inappropriate to the new setting,
6. Local institutional procedures for decision making about software acquisition, distribution, and support are ineffective, or
7. Ownership issues make it difficult for the teacher and his or her school to obtain promising software even if it is appropriate or easily modifiable.

Many examples and comments were given about these particular “hot spots” from the end-user perspective. The general impression coming from the authors is that problems such as the above can be powerful factors in the reduction of the portability potential of a product. The product may not be used at all if even one of the above problems is sufficiently present, leading to an obvious portability failure.

Factors Receiving Little Attention

While most of the factors in the list given in Collis and De Diana (1990b) received comment from the authors, a few did not. Technical issues relating to physical input of data, algorithmic design principles, operating system considerations, and problems relating to networking, ranging from the level of connecting computers to locally available peripherals to more general sense of distribution and maintenance networks, were little mentioned. Also little mentioned were issues relating to cost—the cost of developing, adapting, acquiring, and using software. The article on cost considerations by Oliveira (1990) is of course focused on these issues, and they are also stressed by Murray-Lasso (1990) and discussed by a few of the other authors. What is more notable, however, is that many of the authors did not mention cost considerations at all. Perhaps these considerations are implicit in the articles, but the

strength of the practical impact of cost considerations on educational software portability was not discussed as much as had been expected.

New Factors

Several factors emerged from the articles that had not been explicitly listed in the original set of factors. These include:

1. The importance of field testing and evaluation and of field involvement in each stage of the portability history of a product,
2. The importance of considering portability throughout each phase of the design and development process and strategies for doing this,
3. The specific consideration of adaptability features of the software from the point of view of the end user as well as from that of someone more technically involved in the process of software conversion,
4. The impact of various approaches to project management during design, development, adaptation, and localization activities, and
5. The identification of the user interface as a particularly important intersection point of many of the portability factors.

A characteristic of these factors is that they are multidimensional rather than unidimensional factors. This suggests an addition to the "list of factors" approach given in the overview chapter (Collis & De Diana, 1990b) and to any general model built on this approach (see Collis & De Diana, 1990a). A distinguishing characteristic of educational software is that it embodies the intersection of various sets of factors from each of the four general groups—technical, educational, cultural and social, and organizational—and thus two-, three-, and even four-dimensional factors also are involved. The design process, for example, clearly reflects both educational and technical factors as well as other factors such as those relating to project management and teacher involvement within the design process. Similarly, the processes of field evaluation, revision, and localization involve factors from all four groups. Considerations relative to the user interface of educational software relate to many different factors in the educational, technical, and cultural/social groups. The message emerging from the articles is that any list of factors influencing educational software portability must explicitly include multidimensional factors as well as factors grouped in convenient categories such as "technical" or "educational."

Solution Areas

Just as the authors agree on many common problems affecting educational software portability, there is also some consensus on strategies for the improvement of the portability potential of a software product. These include:

1. Continued improvements in strategies for software design and architecture, with more standardization of approach and technique,
2. More user-appropriate authoring tools,
3. Careful examination of instructional approaches most effective for educational software including the stimulation of high-level dialogue about where computer use can be most educationally effective,
4. More and better teacher involvement within each step of the software life cycle,
5. A greater sensitivity to the pervasiveness of cultural assumptions and other aspects of cultural identity in any educational product,
6. Increasing attention to the development of networks and better strategies for the dissemination and sharing of computer-related educational resources.
7. More efficient and effective management strategies for design, development, and distribution processes.

As these solution areas mature, not only should the portability of educational software improve, but, more fundamentally, so also should its overall quality and effectiveness.

DIFFERENT VIEWPOINTS WITH RESPECT TO THE FACTORS

Although different authors emphasized different subsets of the factors, there was general consensus relative to the overall applicability of the factors and with respect to many of the hot spots and solution areas described above. However, there did appear to be some differences of opinion about certain important overall issues. These include:

1. Should it be more desirable to identify and develop products that are most likely to be usable almost as-is in a variety of educational settings (i.e., content- and culture-free software) with as little a need for local adaptation as possible, or is it more desirable to focus on improving the user-adaptability aspects of software, with the expectation that end users will want to alter, shape, and localize software?
2. Are portability issues most pertinent within the context of resource provision in less-developed countries, or are they also important in relatively resource-rich environments, such as North America and Western Europe?
3. Should an emphasis be placed on advancing technical strategies for the improvement of portability, for example, through more powerful authoring tools, or should the emphasis be placed on identifying more educationally effective applications of computers so that consensus develops on a need and market for certain categories of software products?
4. Is portability a solvable problem, or are the many factors involved and the cost of dealing effectively with those factors too heavy to expect much success?

This special issue will make a useful contribution if it can stimulate debate about these important issues.

AREAS FOR FURTHER RESEARCH AND ATTENTION

In addition to stimulating debate, a synthesis of the contributions in the special issue also indicates key research and policy issues for further investigation.

Research Areas

There is productive research to be done in at least the following areas:

1. Improvements in our understanding of what makes educational software effective in different settings,
2. Improvements in software design and development methodologies, taking into account new authoring and prototyping tools and environments, and moving toward more modularization of software components,
3. Continued work in the development of adaptable software (see De Diana & de Vries, 1990),
4. Developing new procedures for predicting the costs and cost effectiveness of educational software products.

Policy Areas

In addition, areas related to policy and organization have emerged that merit collaborative efforts at the international level. These include:

1. Better strategies for the sharing and distribution of computer-related educational resources, including electronic networks and human networks (such as those nurtured within international professional societies such as the International Society for Technology in Education),
2. More attention to copyright and ownership issues, moving toward some sort of international standardization and a better understanding of the needs and perspectives of both producers and acquirers of software,
3. More attention to strategies for estimating the cost and the cost effectiveness of different types of educational software and different types of development and acquisition processes.

It can be seen that cost and cost effectiveness issues appear in both lists of areas for future work. The cost-effectiveness issue may be the most productive area for collaboration between researchers, practitioners, developers and producers, and policy makers within the whole domain of educational software.