DOCUMENT RESUME

ED 053 565 EM 009 148

AUTHOR Dwyer, Thomas A.

TITLE On the Imortance of Complexity in Supportive Systems for Educational Computing.

PUB DATE 70

NOTE 8p.

EDRS PRICE EDRS Price MF-$0.65 HC-$3.29

DESCRIPTORS *Ancillary Services, *Computer Assisted Instruction, Computer Programs, Curriculum Development, Curriculum Planning, Data Processing, Educational Technology, Individualized Instruction, Inservice Teacher Education, *Interaction, Program Administration, Program Costs, Programing Languages, *Program Planning, Public Relations, Research Tools, Scheduling, Secondary Education

ABSTRACT

To help investigators become more aware of hidden limitations in their work, some important supportive systems to the application of interactive computing in education are described. Support systems are grouped under the headings of technical support, pedagogical and logistical support, and administrative support and are further illustrated by examples under each heading: Technical support includes system-level software, language-level software, and research and development; pedagogical and logistical support is divided into curriculum material support, scheduling, and critical size; and administrative support covers teacher training, public relations, and economics. Within each example, three levels of complexity are given to alert educators to the fact that they are dealing with a variety of constraints and to recommend that they take this into account in their planning and ultimate evaluation. The author draws on his experience with Project Solo, a program concerned with the use of interactive computing within conventional Secondary Schools, to point out that systems of inadequate complexity can interact on a total effort in a negative, but often hidden manner. (SH)

ON THE IMPORTANCE OF COMPLEXITY IN SUPPORTIVE SYSTEMS FOR EDUCATIONAL COMPUTING

Thomas A. Dwyer
Department of Computer Science
University of Pittsburgh

We have all smiled at old film records of the varied attempts at powered flight that took place around the turn of the last century. One of the many constraints the experimenters of this era lived with was an unfavorable horsepower to weight ratio in their power sources. Those experimenters who ignored this constraint (or more likely did not recognize its significance) went their merry way using everything from human powered flapping devices to on-board steam engines. The unhappy consequences of mismatching resources to ideas are well documented in the annals of early aviation.

The application of innovative ideas to educational settings that are part of a formal school "system" is subject to an even wider variety of constraints. Many of these constraints, either because they appear to be peripheral to the innovative project, or because they are too difficult to deal with by the investigators directly connected with the work, are begrudgingly accepted as part of the experiment at hand. Such acceptance can play havoc with some of the most important aspects of a new project, and sometimes the entire project itself.

While an analogy between educational experimentation and an exploration with as mechanistic a goal as powered flight is bound to limp, it provides two useful insights. The first is that the success of the main thrust of an exploration (whether powered flight or computer augmented learning) can be highly dependent on the sophistication of its key supportive systems (whether efficient power sources for the airframe designer, or appropriately complex technology for the educator). The second insight is that a dependency exists between mutually supportive systems, and that the development of these systems is enhanced when their designers interact. If the circa 1900 engine designer were to have developed his product as far as his background suggested, and then frozen that design, the airframe that would have evolved would bear little resemblance to today's Lear jet.

It is more and more the case, of course, that in the world of engineering development there is interaction in the design of supportive systems, and that as a result, the sophistication of these systems eventually meets the requirements for total project success. One of the most impressive characteristics of recent projects (such as the Apollo moon landing program) has been, in fact, the success of the coordination effort expended so that no one sub-system of the total project would, because of its inadequacy, impose a lesser set of achievable goals than had been set for the total project.

One doesn't have to read the works of very many educational philosophers, especially those concerned with human values, to realize that educational goals are not as readily characterized as those of science and engineering. Even if majority agreement could be reached in this area, the educator is still confronted with the marvelous, frustrating, fascinating, bewildering, but always challenging, variability in the object of his work, the human person. Because of variability in students (and educators!), there is a tendency to accept variability in the supportive systems used in exploring new educational ideas, often variability with a range that would make a scientifically oriented experimenter throw up his hands in despair. Although I would be the last to advocate designing educational systems by engineering-like formulae, it seems to me that attempting to innovate with supportive systems that don't begin to match the sophistication of the human learner should be viewed as a betrayal, not a consequence, of a humanistic approach to education.

The purpose of this note is to point out some deficiencies in auxiliary systems that are related to the successful application of interactive computing to education. I am not arguing against simple or low cost systems, nor am I equating high budget with excellence in educational innovation. Some of the most expensive "dedicated" CAI systems in operation are those that I find least interesting. On the other hand, classroom aids, such as paper "computers", or audio-visual materials that "bring the exciting world of data processing into every classroom", should be carefully distinguished from interactive computing that calls on the full power of modern digital systems. The simpler devices are fine; they are valuable tools in the hands of a good teacher. The point is that a complex tool such as interactive computing is not just "finer"--it is radically different, highly dependent on the presence and sophistication of supportive systems ... and capable of transforming the learning process into a form that has no comparable antecedent.

Supportive Systems for Interactive Computing

The suggestions I will make on discerning and building appropriate support systems are based on experiences with an experiment called "Project Solo". This is a program (supported in part by NSF) concerned with the use of interactive computing within conventional secondary schools. We use the term "interactive" to connote (1) a number of technical features as described below under software support, and (2) an educational philosophy that encourages involvement and control by the learner, with the invitation to modify or create programs at all levels, whether they be CAI drills or advanced modeling systems. (cf. reference 1)

The important support systems we have recognized fall under three headings. Under each heading several examples of support that is relevant to interactive educational computing are given. Within each example I have described three levels (mini, midi, and maxi) of complexity. The reason for doing this is not to berate the mini versions, but rather to alert educators to the fact that they are dealing with a variety of constraints, and to recommend that they take this into account in their planning and ultimate evaluation. The items are meant then as a check-list which can give investigators early warning of project "soft" spots, even though it might not be clear what can be immediately done about such weaknesses. (In our own work we spotted 2.c. early enough to gather the resources needed to move toward maxi support; we are still concerned about not getting past the midi level of 2.b.)

1. Technical Support
   a. System-level Software

      Mini: Single compiler, users share core, programs limited in size, no file features.

      Midi: Single (extended) compiler, users swapped; disc files, larger programs, & linking possible.

      Maxi: Multi-Processor, including very extended compilers; sophisticated executive system, extensive file capability, sophisticated time-sharing.

   b. Language-level Software

      Mini: Rudimentary BASIC.

      Midi: BASIC with some extensions, file commands.

      Maxi: Very extended BASIC compiled incrementally to allow JOSS-like interaction; able to handle large business, scientific, or administrative problems as well as CAI programs; (cf. reference 2 for the description of such a language); other processors available (e.g. FORTRAN, ASSEMBLER, SNOBOL, a simulation language, an information retrieval system, etc.).

   c. Research and Development

      Mini: Single users--Very little influence on suppliers of technology to add new features.

      Midi: Cooperative user groups suggest new features to vendors.

      Maxi: Network of users with combined buying power obtain computer services from a source with an experienced research and development staff. (Note: Unfortunately such user power can be lost when regional projects originate with a center that also acts as manager of the project.).

2. Pedagogical and Logistical Support
   a. Curriculum Material Support

      Mini: The local (very busy) teacher produces or collects material.

      Midi: A cooperative regional project circulates ideas from participants and other sources.

      Maxi: Materials are matched to specific curricula; they are produced as a cooperative effort of teachers, subject specialists, and computer specialists; formative evaluation has taken place; complex programs and data are available through the physical devices of a large networkin a form usable by all participants.

b. Logistical Support--Scheduling

Mini: No change in school or teacher scheduling possible.

Midi: Some special consideration is given, but within the framework of a scheduled school.

Maxi: A "free school" exists in which student selection of resources (among which is the computer) complements lectures, discussions, etc.

c. Logistical Support--Critical Size

Mini: Use of computer confined to single teacher, single subject(usually mathematics); inadequate space.

Midi: Several levels within same subject (e.g. algebra, senior mathematics, and honors calculus students) with appropriate space and equipment.

Maxi: Several disciplines, several teachers, with very adequate computer access. (Our experience has been that the cross-fertilization of ideas and mutual support that arises from such an approach is decidedly non-trivial. Generous, well designed physical space also makes a big difference).

3. Administrative Support

a. Teacher Training

Mini: Teachers self-taught.

Midi: In-service training through short programs.

Maxi: Full-time study and involvement (usually through federally funded institutes); follow up visits by professional staff; regional meetings to share ideas.

b. Public Relations; recognition

Mini: Non-existent; possibly negative.

Midi: Positive attitude on part of administration.

Maxi: This level is not recommended at start; it should be reserved for verifiable progress, at which time it can be expressed by allocating more....

c. $$$$$$$

Let X=the total annual budget for a school system with grades 1 through 12. Assume that an increment of C% will be added to that budget to support interactive computing with maxi system and maxi language capability. Our experience is that a useful "formula" for a budget increment for interactive computing is:

$$C = 85*N/X$$

where N is the number of students who will use the computer, assuming 20 45-minute sessions per subject per year for students working in pairs at terminals about half the time, working alone the rest of the time. Three levels of dollar support we would suggest, together with sample figures for a school system that has 12,000 students with an average educational cost of $1,000 per student are as follows:

Annual Budget = X = 12,000*$1,000 = $12,000,000

	INCREMENT IN X	N	EXAMPLE INTERPRETATION
Mini:	C = 0.1%	140	. Students in 4 classes could use computer as above.
Midi:	C = 1.0%	1400	. Every 11th and 12th grade student could use the computer in one course as above.
Maxi:	C = 5.0%	7000	. Every high school student (grades 9-12) could use the computer in two courses at level described above.

Some Final Comments

It has been my intention, by pointing out the constraints within which most interactive educational computing lives, to help the users of such computing become aware of some of the hidden limitations on their work. For example, it is very doubtful if many schools are budgeted at the conservative 0.1% mini level given above. Yet the teachers in such schools may very well be asked to demonstrate that they are making educational breakthroughs.

I have also tried to point out that systems of inadequate complexity can interact on a total effort in a negative,but often hidden manner. An example of this we experienced recently arose in trying to adapt some of the more imaginative Project Solo curriculum ideas to the constraints of mini software. In many cases, it was really not possible. If we had not had the advantage of previously working with maxi software,we probably would not even be aware of the deficiencies that exist in the units that were re-written to work within such a software constraint. On the other hand, some of the "maxi" ideas (which were generated because of the presence of a sophisticated system) did carry over. Thus it is important that at least some projects gain the perspective afforded by high-level supportive systems, and that they communicate their ideas and experiences to others. A good, national-level medium for such communication may very well be one of the most important overall meta-systems that should be added to every project's checklist.

References (Preprints of these are available from the author)

1. Dwyer, T. A. Some Principles for the Human Use of Computers in Education. International Journal of Man-Machine Studies. 3,3 (July 1971).

2. Dwyer, T. A. Teacher/Student-authored CAI Using the NEWBASIC System (to appear in CACM).