Research in Interactive Technologies At the Children's Television Workshop

□ Erik F. Strommen
Glenda L. Revelle

Erik F. Strommen is Director of Research for the Interactive Technologies Division of Children's Television Workshop. He attended Cornell and received his Ph.D. from Rutgers University in developmental psychology and research design in 1988. Glenda L. Revelle is Vice President for Production, Research and Development in Interactive Technologies. She holds a Ph.D. in developmental psychology from the University of Michigan and has been researching and producing software at CTW for seven years.

This article presents a model for integrating research into the software design process, based on the experience of the Interactive Technologies Division of the Children's Television Workshop. The model has three components: (a) using existing research literature to inform initial design decisions and suggest possible issues to be tested during the development of prototype products, (b) conducting in-house basic research studies to provide information unavailable in the published literature, and (c) conducting formative research studies on products in development to assess their usability and to ensure their effectiveness as learning tools. Issues in interpreting published studies are considered, and insights from the results of basic and formative studies conducted by the Interactive Technologies Division are also reviewed.

□ Research results provide vital information to the product-development process. Research helps answer questions concerning the design of products in development, and it provides a starting point for the discussion of innovative products yet to be developed. For those who create and design interactive learning materials, the need for research data is especially intense because the use of interactive technologies in the field of children's education is still relatively new and, as a consequence, the existing literature on effective use is quite thin.

This article presents an overview of the research issues that have confronted the Interactive Technologies Division[1] (ITD) of the Children's Television Workshop (CTW) in the design of interactive learning products. It then proposes a research model for solving design problems that is based on ITD's 12 years of research on children and computers. Under this model, ITD researchers (a) review the basic research literature published in academic and trade journals, (b) conduct in-house basic research on general topics re-

[1]The Interactive Technologies Division (ITD) develops computer software and interactive video products based on the themes, characters, and curricula of CTW's television shows. These products are designed, as are the TV shows, to be educational and entertaining, and to provide quality children's programming. The group creates software for personal computers, video game systems (such as Nintendo), and advanced multimedia systems combining video and audio with computer graphics and interactivity. The ITD works with a variety of outside publishers who currently market more than 20 *Sesame Street*, *The Electric Company*, and 3-2-1 *Contact* titles for use in homes and schools.

lated to children's use of computers, and (c) conduct formative research to ensure the effectiveness and appeal of the interactive features included in new products as they are developed.

We hope our observations about how young children use, and relate to, interactive technologies may be of assistance to others who are interested in developing or using interactive children's products.

INTERACTIVITY AS A RESEARCH TOPIC

One indication of the unsettled nature of the current literature on interactivity is the lack of consensus among researchers as to what the word *interactivity* means. Although the term is used in a wide variety of contexts, it is striking to consider that no specific definition for the term *interactive* has been commonly accepted. Indeed, its pervasive use to describe a wide variety of activities (many of which are not related to any technology at all) suggests that the definition of interactivity may be a function of the medium being described (Jones, 1989; see also the way the term is employed, in relation to print materials, by Link and Cherow-O'Leary elsewhere in this issue).

Before computers became prevalent, the term was often used (in the context of children's products) to indicate materials that in some way encouraged a child to respond or play along at home. For example, in a typical interactive *Sesame Street* segment, Big Bird might look at the camera and ask, "Which one of these shapes is a square? Point to the one that's a square." Whether the child responds or not, the television program proceeds exactly according to the script. The child has no impact on the progress or direction of the program itself.

Is such a situation interactive? For its purposes, ITD prefers a more precise definition. Truly interactive technologies, we believe, are those in which the materials seen, heard, or used by the child are in some way modified on the basis of input from the child. The child must actually be presented with different information or options as a result of responding one way rather than another. For example, an interactive program would be one that asks a child to select a square or a circle and then takes a different course depending on which shape she selects—or, indeed, if she makes no selection at all.

As our definition might suggest, two related issues form the focus of ITD's research activities: the capacity of the child to understand and control an interactive program, and the program's ability to respond to her.

THE ROLE OF RESEARCH LITERATURE

ITD reviews a wide variety of academic, or basic, research studies to obtain information about children and technology. Among the types of research ITD has found most useful are (a) studies of human-computer interactions, (b) educational studies of children and computers, and (c) studies that deal with the cognitive abilities of children and, especially, with their ability to understand and execute specific interactive tasks.

The application of published research results to the solution of product-development problems is not straightforward, however. There are often limits to the utility of the literature in each of these three areas.

Basic Research on Human-Computer Interactions

While there may be no agreed-upon definition of interactivity, there is nonetheless wide agreement among researchers that user-computer interactions are cognitive in nature, and that an effective interactive design must take into account the user's cognitive abilities. Designers must be careful not only about how clearly their product presents information, but also about how clearly their product explains what can be done with that information (Card, Moran, & Newell, 1983; Jones, 1989; Nickerson, 1986). Most studies of user-computer interactions have been written from the perspective of information-processing theory (cf., Bovair, Kieras, & Polson, 1990), focusing on adults performing job-related tasks, such as text editing, programming, or data entry. The adult-oriented, nondevelopmental nature of this literature reduces its value as a resource for designers of materials for young children. As far as we can determine, no studies in this field have attempted to consider the ways in which adults and children may differ in their abilities to use computers. This is a serious weakness in the field—one that we hope will be remedied in coming years.

Yet despite its lack of a developmental focus, the literature on adults and interactivity has been helpful to ITD in several ways.

First, studies of adult users have given us a broad idea of what competent user behavior looks like, and how it is acquired. These studies also help us pinpoint the specific problems that novices of all ages are likely to confront as they learn to operate a new system. For example, some studies indicate that beginning users tend to formulate a general idea of what they want to accomplish with a computer before they actually attempt it (such as "I want to be able to move my cursor back and forth between the top and bottom of a document"). Only afterward do these users try to find the precise commands that will execute that task. Through their analyses of the various ways in which adult users approach and execute interactive tasks, these studies provide us with useful insights into how children might approach analogous tasks, and how those tasks might be broken down into simpler components that children can understand (Card et al., 1983; Riley, 1986).

Producers of interactive materials for children and producers of interactive materials for adults face similar kinds of design problems. These problems range from questions of how best to introduce the novice learner to a new system (cf., Carroll, Smith-Kerker, Ford, & Mazur-Rimetz, 1987–1988) to how best to integrate nonspeech audio into interactive computer materials (cf., Blattner, Sumikawa, & Greenberg, 1989) to how best to provide feedback on errors during system use (Jones, 1989; Norman, 1988). In these examples, the cognitive issues relevant to systems designed for adults are also relevant to systems designed for children, and occasionally the solutions devised for adult-oriented systems can serve as initial models for the solution of problems in child-centered systems.

The adult-user literature can yield a second, even more important, benefit if it is read with a developmental eye. Studies of adult users can flag, or highlight, particular cognitive factors that might be of only passing concern to designers of systems for adults but are likely to be of crucial importance to designers of systems for children. It has been extensively documented, for example, that adult users are quick to develop a *system image*, or mental model, of how an interactive computer task functions based on their previous experience with other instruments, such as a typewriter (cf., Allwood, 1986; Carroll & Rosson, 1987; Norman, 1988; Owen, 1986; Sein & Bostrom, 1989). By analogy, they reason that a computer will operate "like a typewriter" when it is used for word processing. The advantage of mental models is that they allow adult users to organize their expectations about how a task should be performed and what the likely outcomes of their actions will be. The disadvantage of such models is that when the analogy is wrong, errors occur. For example, novice adult users will often treat the ENTER key as if it were a typewriter carriage return, when in fact it does not always serve that function in word processing.

A long line of developmental research has demonstrated that children, like adults, can understand and use analogies (cf., Gardner & Winner, 1986), so there is good reason to believe that children, like adults, could use analogical models of other tasks to guide their computer use. But the evidence also suggests that a child's ability to reason analogically is limited. It depends on how similar the two objects (or tasks) being analogized may seem to the child. It also depends on how simply and concretely the analogies are presented (Eson & Cometa, 1978; Winner, Engel, & Gardner, 1980). Thus, although the research literature on adult users and mental models is suggestive, its findings cannot be applied without qualification to the design of interactive materials for children.

Basic Educational Research on Children and Computers

There is a burgeoning educational literature on children and computers, as illustrated by the appearance of a number of new journals devoted to the topic (*Computers in the Schools*, *Journal of Educational Computing Research*, *Journal of Computing in Childhood Education*, and *Interactive Learning Environments*, to name a few). Paradoxically, this literature has actually been less helpful for ITD's specific purposes than the human-computer-interaction literature cited above. This is because educational research on children and computers has tended to focus on issues relating to the instructional role of computers in schools—such issues as whether computers can be used effectively to teach specific curricular content (cf., Deatsman & Keough, 1989) and how learning takes place on classroom computer networks (cf., Scardamalia, Bereiter, McLean, Swallow, & Woodruff, 1989). Although these studies can be useful in designing the content of interactive learning materials, they are not especially enlightening when it comes to many of the functional problems involved in designing new software. Such issues as how well a specific computer task is understood by children, how easy a software program is to manipulate, and what aspects of a given program are most problematic for children are not usually addressed in this literature.

Nonetheless, the educational literature is useful because it is often possible to read between the lines of these studies to find valuable information. Shade, Nida, Lipinski, and Watson's (1986) study of computers in a preschool class, for example, was designed as a descriptive account of the computer's role in the classroom, but it also provides information on young children's attitudes toward computers, their use of the keyboard, and their grasp of the software. Thus, while educators clearly have a different research agenda than designers of interactive learning media, their studies are still of interest in both a general and sometimes more pragmatic way.

Basic Developmental Research

The basic research literature cited thus far—studies of human-computer interactions and the educational use of computers—has been specifically related to computers. Perhaps the most important area of research for ITD, though, is one that usually does not involve computers: basic developmental research on children's physical and cognitive abilities. This research contributes to effective product development in two ways.

First, when one designs a learning task for children of a given age, it is crucial to know the general level of skills children may be bringing to the task and how able they may be to attack problems relating to the subject matter being taught. If one is designing a music-teaching program, for example, it would be helpful to review the available literature to find out what musical concepts, such as pitch or meter, children can—or cannot—grasp at a given age (cf., Jones, 1976; Van Zee, 1976). It also would be useful to know how young children graphically represent these concepts (cf., Goodnow, 1971; Hildebrandt, 1987).

The discovery, for example, that young children can discriminate and graphically portray differences between high and low notes, but not between long and short notes, can provide a starting point for the initial design. Similarly, if one is designing a game that teaches simple addition and subtraction, knowledge of how children use intuition and counting strategies to solve such problems (cf., Brush, 1978; Carpenter & Moser, 1984) can help the designer create a program that capitalizes on children's inherent skills.

A second way in which the existing literature can be useful is as a source of normative information on both the nature and the limits of children's cognitive abilities. The knowledge that children have smaller working memories than adults (Case, 1985), for example, should serve as a warning to designers not to devise tasks for children that have too many levels of complexity or present them with too many options. Similarly, the fact that children have comparatively poor visual discrimination (Enns & Cameron, 1987; Enns & Girgus, 1985) has implications for the design of computer screens. To cite another example, ITD is particularly interested in children's comprehension of verbally presented material (see below). As speech becomes an increasingly common feature of interactive technologies, it becomes very important for software designers to know how easily children can interpret and respond to spoken information. In the near future, at least, the typical affordable computer will be able to repeat only a limited number of stock phrases. As a result, questions such as how long should these phrases be, and how should they be worded, assume vital importance.

THE UTILITY OF IN-HOUSE BASIC RESEARCH

Because the existing literature is not completely adequate for ITD's purposes, the division conducts its own basic research. One interesting area ITD has studied is interface design. The term *interface* refers to those aspects of the computer hardware and software that enable the user to control the program. Interface design involves hardware issues (e.g., what kind of physical device, such as joystick or mouse, best allows the child to interact with the computer) and software issues (e.g., what are the program's mechanisms for allowing the child to make choices, register responses, and perform activities).

The distinction between these two aspects of the interface is somewhat artificial. The choice of hardware device, for example, often has a direct impact on the selection of software interface (such as the type of menu used). For the purposes of this analysis, however, we will maintain the artificial distinction between the two aspects of the interface.

Input Devices

The hardware interface is by no means a minor aspect of computer control. The physical demands of using different devices, such as a mouse or a keyboard, can have a powerful effect on adult performance with computers (Card, English, & Burr, 1978; Card et al., 1983; Ewing, Mehrabanzad, Sheck, Ostroff, & Shneiderman, 1986; Karat, McDonald, & Anderson, 1986; Whitfield, Ball, & Bird, 1983), an effect that may be amplified for young children. Wallace, Newell, and Wade (1978) have demonstrated that young children have more difficulty coordinating hand-eye activities than do adults. This suggests that young children would also have more problems manipulating an input device. Interface use involves more than hand-eye coordination, though. Users must be able to monitor the results of their actions on a computer screen (which in most cases is physically separate from the input device), and they must be able to understand the rules by which particular input devices operate.

These demands raise significant questions about whether young children are able to use computers as they are currently designed. Formative studies on software development have indicated that standard computer input mechanisms may present cognitive and/or motor demands that strain the abilities of many preschoolers (Char, 1990; Offerman, 1989). Children can spend so much time and effort attempting to master the basic interface that the educational content of the program becomes inaccessible. Failure to consider the problems of input-device use and how they might relate to the content demands of children's software would be a serious oversight.

When we at CTW began investigating interface issues, we discovered that although the existing literature on children's use of different input devices was sparse, it clearly suggested that most currently available interfaces presented unique problems for young children. Grover (1986), for example, found that both preschoolers and kindergartners spent a considerable amount of time scanning back and forth between the screen and keyboard in a frustrating search for keys. In addition, they frequently made errors, striking the key adjacent to the one intended or mentally confusing U with V, Y with U, and M with N. Shade et al. (1986) also reported that the children in their sample had trouble using the keyboard, occasionally pressing multiple keys together in what the authors dubbed the "piano playing" technique.

With respect to other input devices, Avons, Beveridge, Hickman, and Hitch (1983) found that although both 5-year-olds and 10-year-olds were able to use a light pen, the 5-year-olds were less accurate in their placements on the screen, which tended to drift off the target being selected. Five- and 6-year-olds' use of knob-based paddles for playing a computer game was described by Silvern, Williamson, and Countermine (1988) in their study of spontaneous computer use. They observed that the children frequently failed to notice when the paddle they thought they were "using" was in fact not under their control—a phenomenon analogous to young children "playing" video games in arcades when the machines are actually running in demonstration mode. Cunningham (1985) did report that children as young as 3 could use a touchpad and stylus for guiding a cursor across the computer screen in a simple target-seeking task; however, no details of their performance were provided, so it is difficult to evaluate this claim.

The inconclusive nature of these studies for our purposes presented us with a dilemma. We wanted to design materials that would be easy for young children to use, but no comparative studies of the different devices existed. This led us to conduct our own basic research aimed at determining which device worked best, assuming standardized conditions. In two studies (Revelle & Strommen, in press; Revelle, Strommen, & Offerman, 1990), we compared the performances of 3- and 4-year-olds using keyboard arrow keys, a light pen, a touch screen, a joystick, a trackball, and a mouse. We found that the easiest devices for children to use were the touch screen and the light pen, because they allowed children to simply point to or touch their desired choices (see Char, 1990, for important practical problems in the use of the touch screen, however). The next best devices were the trackball and the mouse. The joystick and the arrow keys were the most difficult and frustrating devices for children to use.

Our provisional explanation for these results can be found in Revelle and Strommen (in press) and Revelle et al. (1990). The important point to be made here is that ITD uses basic research results in a very directed way. Although we do develop "futuristic" or new technology products (such as interactive compact discs, described later), our primary emphasis is on developing educational products for technologies that are currently available in homes and schools, or are commercially available at a reasonable cost to most consumers. So even though light pens and touch screens seem to be ideal input devices for children, their current high cost and limited availability make them less than ideal choices for use with our software. Alternatively, because young children have difficulty using the more readily available joysticks and arrow keys, we tend to avoid using them as input devices. We have tried instead to design our software for use with a mouse or a trackball, which are easy for children to use, only slightly less popular than joysticks, and widely available as computer input devices.

Through our basic research, we have also been able to apply our findings to other input devices that share the same properties as the ones we have tested. For example, we have developed three Nintendo game cartridges (Children's Television Workshop [CTW], 1988, 1989a, 1990), all of them designed to be used with the conventional Nintendo controller. The Nintendo controller uses a single black, cross-shaped button to move the cursor. Each arm of the cross is labeled with a directional arrow, with each arrow corresponding to a cursor movement (up, down, left, or right).

Because our previous research had indicated that arrow keys are difficult for children to use, we wondered how easily young children would be able to play these games. This led us to modify the software in a way that took into account any potential interface problems with the Nintendo button. Specifically, we devised a system of "cycle-and-choose" options, so that any press of the button moved the cursor to the next menu option in a left-to-right sequence. As a result, we were able to simplify the child-user's task immensely. We are currently conducting a study of how young children use the Nintendo controller in order to determine (a) if there are any consistent strategies children use with the controller that we might capitalize on, and (b) at what age children are able to employ the directional arrows on the Nintendo controller effectively.

Software Factors

It has been suggested that graphics-based interfaces are the most appropriate and easiest for young chidren to use (Porter, Lahm, Behrmann, & Collins, 1986; Wright, Shade, Thouvenelle, & Davidson, 1989). By and large, young children have not yet learned how to read. Consequently, they are usually unable to respond to text messages on the screen or to type in written commands. In graphics-based interfaces, small pictures, or *icons*, representing response options appear on the screen, replacing or supplementing text. Computer control is achieved by moving a cursor to the appropriate icon, and then pressing a button to select the option it represents. Thus the child's ability to manipulate the input device so as to control the cursor figures prominently in graphics-based interfaces, because program use proceeds through a series of "move cursor to icon, press button" steps.

In our study of six input devices described earlier (Revelle et al., 1990), we examined the effectiveness of the input devices under two cursor conditions. In the first condition, a frame cursor simply leapt from one icon to another; in the second, we used an arrow cursor similar to those used with most mouse-based interfaces. Because this latter cursor could move all over the screen, it had to be manually positioned on the desired icon in order to register a response. The results indicated that certain device-cursor combinations are easier for young children to use than others (interested readers are referred to the actual paper for details). A second study, currently in progress, is comparing children's performance with three different input devices—a mouse, a joystick, and a trackball—under varying screen conditions. In this study, children are using the devices with icons of different size, and screen fields that vary in size and complexity. From this study, we hope to gain insight into how performance with different input devices improves with practice, and how different screen designs might enhance or inhibit children's ability to use the different devices.

ITD has studied children's ability to understand different screen environments. The main issues we have considered in these studies are the form, size, and location of icons on the screen. In general, we have found that children interpret icons very concretely, and that the most effective icons are those that depict their functions most directly. For example, a "door" icon that allows the user to exit a screen has proved very effective (Strommen, in progress-a). Most recently, we have found that when children use drawing software that features a "paintbrush" cursor for filling in areas with color, they try to "paint" with the cursor. They move the cursor with the same back-and-forth strokes they might use with a real brush on a piece of paper, rather than positioning it over an area and pressing the button to fill in the color—which is the protocol required by most software (Strommen, in progress-b).

Two other important issues in icon design are the size of the active area, or *hot spot*, associated with the icon, and the amount of separation between icons on the screen. When young children move a cursor to an icon, for example, they frequently overshoot their mark (or stop short of it, for fear that they might overshoot it). We have found that icons having small or narrow hot spots can be frustrating to children, who end up spending too much energy on controlling the cursor (Strommen, 1990a, in progress-a). We have found that putting a large active area around each icon is the best solution. Unfortunately, this solution can create additional complications. If icons are placed too close together on the screen so that their hot spots border one another, children are more likely to accidentally select the wrong icon than they are if the icons are more widely separated (Strommen, 1990b). Children seem to fail to notice that the cursor is not positioned correctly before they press the button to make a selection. Obviously, if the cursor is positioned over empty space, nothing happens. But if it ends up resting on another icon's hot spot, children wind up selecting something they did not anticipate or want. Given these sorts of problems, we often find ourselves having to make tradeoffs in our designs between icons having big hot spots (for easy targeting) and icons with wide separations between them (to minimize the risk of error).

THE NEED FOR FORMATIVE RESEARCH

While in-house basic research is essential to ITD's work, we find we cannot rely on basic research alone to ensure product effectiveness. It is impossible to create optimal interactive materials without doing a careful analysis of how a particular product performs with actual users in real-life contexts. Flagg (1990) has argued that instructional materials developed solely on designer intuitions are consistently inferior to those developed with feedback from actual users during product development. The methods for obtaining this feedback have come to be known as *formative evaluations* or *formative research*—the term *formative* referring to the fact that the product being tested is still being developed, or formed, while the studies are being conducted (Flagg, 1990). As described by Flagg (1990) and Schauble (1990), formative research involves the pretesting of proposed educational or other materials to ensure that they can achieve their desired goals. Such testing usually involves the creation of a *prototype,* or rough model, of the actual materials to be used.

At ITD, one of our staff members is a programmer whose main job is to develop prototypes for testing with children. We cannot stress strongly enough how important we have found formative evaluations to be in developing our products. If a child cannot understand a segment on an educational television program, she can wait for the next segment to come on. In contrast, young users of our interactive software must be able to understand how to use our products in order for our products to be effective. In the field of interactive technologies, in particular, formative testing can provide crucial feedback on the strengths and weaknesses of particular materials and also can allow for the modification of problematic features prior to the release of the final product.

Formative research serves another, invaluable purpose: It provides a corpus of documented in-house knowledge based on past experience. As studies of previously developed products accumulate, they form an indispensable resource for anticipating problems with new products still in the conceptual stages of development. Comments such as "We tried that with product X, and the children couldn't understand it, remember?" can steer designers away from areas already known to be problematic. Conversely, those elements previously shown to be effective can be used or modified in the development of subsequent interactive products. First-hand knowledge of what works and what doesn't means not having to "reinvent the wheel" with each new product, and it ultimately leads to a streamlined design process and better products.

Only the actual testing of a product can reveal whether it works as expected, and whether users enjoy it and find it beneficial. One problem confronting designers of interactive materials in different media is that the levels of interactivity in different technologies are very different. Each technology has features that differentiate it from others, and these features have a direct impact not only on how much control users have, but also on how effectively users can exercise that control. In the next sections, we discuss examples of how research has helped us answer the different questions that have come up when we have developed educational materials in different media. These descriptions should not be taken as definitive analyses of the important properties of the different technologies discussed. Rather, they reflect the issues that arose in the creation of a given application, and how we tried to resolve them both by referring to the existing literature and by collecting data on children's actual performances under varying conditions.

Computer Software

At CTW, research questions concerning computer software for children usually revolve around problems associated with user control. By this we mean the child's ability to invoke any of several different program functions and to use the functions with minimal instruction. All of the functions available in a given program must be readily comprehensible to a child, and the child-user must be able both to understand the results of a given action and to act on them with minimal effort. In word processing, for example, there are many complicated options to choose from: you can type text, move the cursor around on a page, move to different pages in a document, change fonts, change margins, and so forth. How can a child grasp all these options?

These issues came to the fore when we conducted research on *Sesame Street First Writer*, our own word processor for preschoolers (CTW, 1989b). The concept behind *First Writer* was to create a simplified text editor that could be used by an adult along with a small child just learning about letters, words, and writing. Aware of the tendency of young children to use concrete analogies to help them understand computer tasks (noted earlier), we designed a product that drew on, as much as possible, children's experiences with writing.

The initial prototype of *First Writer* differed from conventional word processors in three main ways. First, it did not permit the user to scroll through pages. When the end of a page was reached, the page had to be advanced by pressing the down-arrow key—just as in letter writing one has to turn to a brand new page after the current page is filled. The shift to a new page was marked by the changing of a visible page number, and by an animated sequence that showed a page of writing curling away, revealing a fresh, blank page underneath.

The second distinctive feature of *First Writer* was its use of large letters in fonts that were linked by color and shape to specific *Sesame Street* characters. Used for writing text, these letters produced pleasing felt tip marker-like images. An icon of a *Sesame Street* character, displayed in one corner of the screen, indicated which font was in use. Pressing a function key changed the face, and thus the font (Big Bird's font consisted of tall, yellow letters; Cookie Monster's letters were fuzzy and blue, etc.).

Finally, *First Writer* had no *word-wrapping* capability. (Word wrapping involves the automatic shift of a whole word that does not fit at the end of a line to the beginning of the line below.) Children had to press the return key to advance to the next line when they ran out of space. If children tried to insert letters in an already-completed line, an animation (bricks crunching together in a wall) appeared and a beep sounded, signaling the lack of space on the line. To position the cursor on different lines without actually typing, children had to press the return key or use the down-arrow key.

The prototype program was tested with a sample of forty 4-year-old children (Offerman, 1989). With respect to its features, we found that although more than half the children were able to use the page-turning function after only one short session of use (and showed signs of continued improvement), fewer than 25% understood the graphic image of the turning page, and most found it confusing. The different letter fonts worked well: After some initial confusion, the majority of children could use them very easily, and could select the ones they wanted. The elimination of automatic word wrapping, however, made writing more cumbersome. It added an extra step to using the software, as the children repeatedly searched for the return key or forgot how to use it. But the most problematic of our features was the "brick wall" animation, indicating that a line was full. None of the children understood what it was or why it was there. These results were grist for our design team, providing it with concrete feedback about the effectiveness of each proposed feature.

One other interesting finding confirmed an observation of ours: More than half the children tended to hold down the keys they pressed, at times producing text with multiple letters ("Berrrrrrrt" instead of "Bert," for example), at other times causing a rapid (and unintended) advance of lines and pages when they held down the arrow keys.

How to apply these results?

Because the program was designed to be used along with an adult, we judged some of its basic features to be functional as they were. The fonts presented no problems, and were kept unchanged. Because children understood how to turn a page, even though they did not immediately grasp the meaning of the "turning page" animation, we "tweaked" it but did not alter it substantially. We also eliminated the need to press the return key at the end of each line, and changed the product so that it automatically continued displaying letters on the next line as they were typed. This new feature proved much easier for children to use. For the brick wall animation, we substituted an animation of a *Sesame Street* character shaking his head, to gently suggest to children that something about their previous action was in error (nodding- and shaking-head animations worked very successfully in one of our other products, *Big Bird's Special Delivery* [CTW, 1987]).

There still remained, of course, the problem of the children holding down the keys. Although we had gotten a few hints of this problem from our work on previous products, the issue of a child's effective use of the keyboard had never been a central issue. (Our previous work tended to focus on either single-key responses or the use of other input devices, such as a mouse or trackball.) We have since come to suspect that the problem may be a manifestation of a more general developmental problem of children, and we plan to conduct some basic research to determine if this is true (Strommen, 1989c).

In developing *First Writer*, however, we had to deal directly with the issue of effective use. We ended up solving this problem very simply: The program now disables the *autorepeat* function on the keyboard. In other words, children can hold down a key as long as they like, and they will not produce repeated letters or repeated cursor movements.

This example illustrates how research helped us improve or modify specific features of our new word-processing software. First, we designed the tasks so that children could understand them in terms of something within their own experience: the act of handwriting. Second, we identified the features of the product that might be problematic, developed a prototype that embodied these features, and tested the features with children to determine if our concerns were valid. Third, we modified the final version of the product (wherever possible) on the basis of the results of this testing.

In general, we make our decisions to modify a product on pragmatic as well as principled grounds. It is simply not feasible to adopt rigid criteria such as "if 50% of the children can't use X, it must be changed," because production deadlines and design constraints often limit our ability to change the features of a program. Thus, we try to fix the problem as best we can, within the constraints of our all-too-limited budgets and production timetables.

Interactive Compact Discs

An emerging technology similar to, but more powerful than, computer software is interactive compact discs. Interactive compact disc software, driven by the same kind of technology as computers, presents many of the same design problems as computer software. Compact discs can store substantially more information than conventional computer discs, however, and this gives designers greater flexibility in solving those problems through the use of much more elaborate graphics and sound effects—especially speech.

In designing our compact disc products, we seek to make good use of the technology's speech capabilities, and our comments in this section address what we have learned about the use of spoken language in systems for young children.

A review of the available literature on children's ability to understand the spoken word reveals what every parent knows: Young children are not particularly good listeners. Listening is a complex cognitive process, and children are poor listeners in several ways. Young children are not attuned to inconsistencies in speech; nor are they good at monitoring their understanding of speech. As a result, they frequently fail to "realize that they don't understand," as Markman (1979) put it (see also Beal & Belarad, 1990). They also have difficulty processing complex sentences in the passive voice, or compound sentences containing prepositions (Hargrove & Panagos, 1982; Tomasello, 1987), and tend to confuse quantitative terms, such as *more* and *less* (Kingma & Loth, 1984).

When we began designing our compact discs, we initially tried to address these problems by making sure the scripts for our products were written in the simple language of characters from *Sesame Street* (who then actually recorded them). The initial instructions to the user and all error messages, prompts, and end-of-game comments were also recorded.

In our formative testing, however, we discovered that young children had greater difficulties following interactive speech than they did following speech on television. As a result, we found that simplifying our script language was not sufficient.

In prototype tests of four of our interactive games with more than eighty 3- to 5-year-olds (Strommen, 1989a, 1989b, 1990a, 1990b), we found four recurring results having implications for speech in interactive products. First, we found that many of the speech segments we had recorded were too long for children. When children watch a television program, they sit and listen; if they lose interest, they wander away or lose attention until something catches their attention again. In the case of our compact disc products, however, the children did not want to sit and watch. They wanted to act. As we explained to our design team, "Put a controller in a child's hand and he doesn't think it's TV, he thinks it's Nintendo." The children sat impatiently, fidgeting or pressing buttons on the controller, when the system "spoke" to them. They found the lengthy, TV-like dialogue frustrating rather than entertaining. To make matters worse, we discovered that the more the children played with the system, the more annoying they found the long strings of dialogue. In short, the children didn't want to have to pause to listen. They wanted to "get on with it."

Second, we found that the children tended to ignore, or miss, important content in a long stream of speech. In our classification game, for example, Ernie hides Bert's paperclip collection in their apartment. He gives the child clues such as "I hid one behind everything that's blue, everything that has wheels, and everything that plays music." This is crucial information for playing the game. Unfortunately, it was embedded in a nearly minute-long segment of dialogue between Ernie and Bert. Although adults had no trouble picking out the important information, children often missed it entirely (Strommen, 1990b).

Third, we found that the children often responded more to the inflection of the speech than to the content of what was said. For example, in one game, designed to teach the spatial locative concepts *near* and *far*, Big Bird hides his birdseed and the child must find it by following Big Bird's clues: "You're very near," "You're kind of far," and so forth. Unfortunately, Big Bird's comment when the child is farthest from locating the hidden object ("You're as far away as you can get!") is one of his most emphatic. Responding to Big Bird's tone of voice rather than his words, children often felt impelled to select the wrong hiding places.

Our fourth finding was a positive one. In one of our games, the children must match a set of objects with the number representing its quantity. For example, given a set of six objects the child must select a "6" from an array of numbers on the screen. We found that many children knew how many objects there were, but did not know what a *six* looked like. By adding a prompt that named the number the cursor had stopped on, we were able to make the game easier to play for those children who were still refining their number-numeral associations.

As a result of these studies, we spent considerable effort editing down the recorded dialogue and adding segments that identified objects, where needed. We streamlined all the messages, shooting for instructions less than 20 seconds long and error messages and prompts less than 10 seconds long. From an adult's perspective, the resulting speech may have lost some of its original character. Many asides, jokes, and tangential comments had to be cut to keep the speech concrete and strictly related to the interactive task. However, these changes shortened the time a child had to sit and listen, and improved the child's ability to understand the rules of the games. By these criteria, we judged them a success.

Interactive Videotape

One feature of both computer software and interactive compact discs is that both are random-access technologies in which the timing and execution of activities are usually dependent on the user. The child can take as long to search for a key as he likes, or can pause to consider where to move the cursor. This feature of the software is often important, because it allows each child to work at his or her own speed. Unfortunately, this self-pacing feature is not characteristic of all interactive media. Specifically, a number of interactive videotape-based systems have been developed that do not allow users to control the pacing of activities. To describe how pacing works in this medium, and how we have used research to address this issue, it is necessary to describe briefly the nature of the technology itself.

The typical computer technology allows for the open-ended branching of task actions: Each of the child's choices gives rise to new choices, with each choice being contingent on what came before. In videotape-based systems, however, the sequence of events is not open-ended, but *linear*—that is, the action proceeds from beginning to end at its own speed, regardless of the child's pace.

Within such a format, opportunities for interaction are restricted—the child can have only a limited impact. In four programs we developed for the View-Master Ideal Interactive Vision system (View-Master Ideal, 1988), we were able to produce interactivity by hooking up a computer with a graphics generator and an audio controller between the VCR and the television set. Interactive videotapes were produced with multiple audio tracks over a single visual image, and these different audio tracks, in turn, were coordinated with interaction-specific graphic overlays via the computer.

The interactivity of such a product is qualitatively different from the interactivity of a computer. User control can occur only at specific points in the tape's narrative. At these points, the child must make choices that dictate which audio-track and graphics branch the story line will follow. For example, after watching a brief introduction, a child may be presented with an interaction in which she gets to select the "number of the day." If she selects 5 as the number of the day, all subsequent interactions will focus on the number 5. As the child watches segments similar to those seen on *Sesame Street* (segments, for example, that deal with the numbers 1 through 10), she is instructed to press a button whenever she sees a 5. If she is correct, she is rewarded with a brief graphic of a 5 and a congratulatory audio message. When she goes back to the beginning and selects another number, however, the audio and graphic overlays keyed to that number would be played. Thus, although the basic visual track remains the same, the game seems different each time because of the choices the child makes. The types of interactions possible in such a system are surprisingly varied. Children can choose different letters and numbers, make words, answer multiple-choice questions, and so forth. It is the child's ability to vary her responses at each choice-point that makes the game seem fresh each time.

Many of the design issues pertaining to computer- or compact-disc-based systems apply to videotape-based systems as well. We still have to be concerned with such issues as what input device we use, how information is displayed on the screen, and how users get feedback. Where this technology differs from the others is its linear pacing. The child must be cued to act at particular points in the tape, and then must do so within a specific time frame, or *window*, as the tape plays. Our research questions concerning interactive videotape have centered on this aspect of the technology, which we have loosely termed the "time marches on" factor. Because the videotape is playing continuously, the window during which an interaction must be executed is of finite duration. But how short can it be?

To answer this question, we undertook a series of studies designed to assess the length of time children took to respond to different types of interactive tasks. The actual research was straightforward. Prototypes of each of the different activities planned for the product were tested with a total of eighty-five 3- to 5-year-old children. We recorded the length of time the children took to begin interacting after prompting, and the length of time they took to complete the interaction—to make a choice, make words, clear the screen of objects, and so forth (Henriquez, 1988).

As it turned out, the task of collecting the data was not nearly as difficult as deciding how to interpret it. We found that the age of the children did not seem to be a major factor in speed of performance. But we also found that different types of interactions took different lengths of time to execute. From a design perspective, this was good: The length of time that had to be allotted for each interaction could be varied according to the observed playing times of the children. But what was the best figure to use in selecting the optimal window for each interaction?

We initially considered taking the median time—that is, the length of time it took 50% of the sample to complete the interaction—as our measure, but this struck many members of our team as too conservative. Was it acceptable for half our users to have time run out on them? On the other hand, if the interval was long enough to accommodate the slowest children, the fastest children might find the long wait before resumption of action very frustrating. Finally, we settled on the 75th percentile as a useful compromise. In other words, all but the slowest 25% of children would be able to finish the task in this time frame, and the delay before the tape continued was acceptable. We used these times as our production standards when we created the final product.

CONCLUSION

The purpose of this paper has been twofold: to offer a model of how research can be used in the development of interactive products, and to demonstrate how we at CTW use this model in our work. At bottom, we place a high priority on research for a very simple reason. It provides us with information on which to base design decisions. The kinds of information we obtain from each type of research—whether basic or formative—serve different purposes, however.

In closing, it might be helpful to clarify the differences between the two types of research and how we use each. Basic research is (a) designed to address particular theoretical questions in various areas, and (b) generally composed of results collected under standardized conditions using traditional experimental procedures. The conclusions of these types of studies tend to provide general answers or results that have a reasonable degree of universality. We can thus rely on them to give us a broad view of a given topic. The research literature on computers, for example, helps us understand the pragmatic requirements of interactive tasks per se, and helps us figure out the best ways to execute them. The developmental literature helps us consider the unique problems of designing materials for the child as user. Judicious use of the developmental literature has helped us take children's special needs into account, and has made us sensitive to what the content of our interactive tasks should be, and how those tasks should be structured.

The basic research literature is frequently inadequate for our purposes, however. When this is the case, we have found it necessary to undertake our own basic research to gain answers to the specific questions we have. These studies are often laborious to design and execute, but the results provide us with well-grounded data on issues that relate to children's use of computers, in general. Our basic research on interfaces, for example, has been indispensable to us. Producing and publishing our own basic research also provides us with an additional—and unexpected— benefit. It helps us maintain contact with researchers at other institutions, both public and private, who are working on similar problems. Through these contacts, a productive exchange of ideas can take place—a dialogue that gives us a window onto new ideas and new methods, and stimulates our own thinking about these issues.

The general information provided by basic research is ultimately of only limited usefulness to us in the daily process of product testing. Each of the individual products we create is unique. How children will respond to them cannot always be predicted accurately. As the examples of formative research described earlier illustrate, aspects of software functioning that seem quite sensible to adults can be problematic for children in unexpected ways, or can require special rethinking when children are the primary users. Formative research, concerned as it is with the concrete aspects of product design, is meant to address these sorts of questions. The information formative research provides is thus much more concrete, and often evaluative in character. If basic research asks the question "What does the ability to do X consist of?", our formative research asks much simpler questions such as "Does product X work?" or "Can children use feature X?"

The answers to these questions do not depend on particular theoretical notions, nor do they have any special generalizability beyond the given product and products similar to it. Formative research, then, provides evaluative information about materials prior to final production, so that problems can be discovered and remedied. These remedies are often ad hoc, rather than systematically constructed; yet because they are made on the basis of evidence from actual performances, they virtually always result in a better product.

In an era of rapidly changing technologies, each having its own special characteristics and often lacking any research base from which children's performance can be predicted, formative testing is a necessity if we expect to produce interactive materials appropriate to young children's needs.

In the Interactive Technologies Division at CTW, we have found both forms of research indispensable. Basic research gives us the conceptual framework from which to work in designing interactive materials for children, and formative research allows us to put our ideas to the test. Through the interplay of formal, basic research and pragmatic, formative research, we feel it is possible to create better interactive materials than we could produce without the benefit of such information. In that spirit, we hope that this paper provides useful information to others who design and use interactive learning materials for children. □

The authors thank Rob Madell for his thoughtful comments on earlier drafts of this article.

REFERENCES

Allwood, C. M. (1986). Novices on the computer: A review of the literature. *International Journal of Man-Machine Studies, 25*, 633–658.

Avons, S. E., Beveridge, M. C., Hickman, A. T., & Hitch, G. J. (1983). Considerations on using a light pen-interactive system with young children. *Behavior Research Methods & Instrumentation, 15*(1), 75–78.

Beal, C. R., & Belarad, S. C. (1990). The development of message evaluation skills in young children. *Child Development, 61*, 705–712.

Blattner, M. M., Sumikawa, D. A., & Greenberg, R. M. (1989). Earcons and icons: Their structure and common design principles. *Human-Computer Interaction, 4*, 11–44.

Bovair, S., Kieras, D. E., & Polson, P. G. (1990). The acquisition and performance of text-editing skill: A cognitive complexity analysis. *Human-Computer Interaction, 5*(1), 1–48.

Brush, L. R. (1978). Preschool children's knowledge of addition and subtraction. *Journal for Research in Mathematics Education, 9*, 44–55.

Card, S. K., English, W. K., & Burr, B. J. (1978). Evaluation of mouse, rate-controlled isometric joystick, step keys, and text keys for text selection on a CRT. *Ergonomics, 21*(8), 601–613.

Card, S. K., Moran, T. P., & Newell, A. (1983). *The psychology of human-computer interaction*. Hillsdale, NJ: Lawrence Erlbaum Associates.

Carpenter, T. P., & Moser, J. M. (1984). The acquisition of addition and subtraction concepts in grades one through three. *Journal for Research in Mathematics Education, 15*, 179–202.

Carroll, J. M., & Rosson, M. B. (1987). Paradox of the active user. In J. M. Carroll (Ed.), *Interfacing thought: Cognitive aspects of human-computer interaction* (pp. 80–111). Cambridge, MA: MIT Press.

Carroll, J. M., Smith-Kerker, P. L., Ford, J. R., & Mazur-Rimetz, S. A. (1987–1988). The minimal manual. *Human-Computer Interaction, 3*, 123–154.

Case, R. (1985). *Intellectual development from birth to adulthood*. Orlando, FL: Academic Press.

Char, C. (1990, April). *Touch, click, or jump? The relative merits of different input devices for use by young children*. Paper presented at the annual meeting of the American Educational Research Association, Boston.

Children's Television Workshop. (1987). *Big Bird's Special Delivery* [Computer program]. New York: Hi Tech Expressions.

Children's Television Workshop. (1988). Sesame Street *123* [Nintendo game cartridge]. New York: Hi Tech Expressions.

Children's Television Workshop. (1989a). Sesame Street *ABC* [Nintendo game cartridge]. New York: Hi Tech Expressions.

Children's Television Workshop. (1989b). Sesame Street *First Writer* [Computer program]. New York: Hi Tech Expressions.

Children's Television Workshop. (1990). *Big Bird's Hide and Speak* [Nintendo game cartridge]. New York: Hi Tech Expressions.

Cunningham, H. A. (1985). An Apple microcomputer-based laboratory system for the study of visual-motor behavior. *Behavior Research Methods, Instruments, and Computers, 17*(4), 484–488.

Deatsman, G. A., & Keough, S. (1989). The effect of computer simulated number blocks on retention of double-digit addition and subtraction ability. *Journal of Computing in Childhood Education, 1*(1), 85–93.

Enns, J. T., & Cameron, S. (1987). Selective attention in young children: The relations between visual search, filtering, and priming. *Journal of Experimental Child Psychology, 44*, 38–63.

Enns, J. T., & Girgus, J. S. (1985). Developmental changes in selective and integrative visual attention. *Journal of Experimental Child Psychology, 40*, 319–337.

Eson, M., & Cometa, M. (1978). Logical operations and metaphor interpretation: A Piagetian perspective. *Child Development, 49*, 649–659.

Ewing, J., Mehrabanzad, S., Sheck, S., Ostroff, D., & Shneiderman, B. (1986). An experimental comparison of a mouse and arrow-jump keys for an interactive encyclopedia. *International Journal of Man-Machine Studies, 24*, 29–45.

Flagg, B. N. (1990). Formative evaluation in decision making. In B. N. Flagg (Ed.), *Formative evaluation for educational technologies* (pp. 13–31). Hillsdale, NJ: Lawrence Erlbaum Associates.

Gardner, H., & Winner, E. (1986). Attitudes and attributes: Children's understanding of metaphor and sarcasm. In M. Perlmutter (Ed.), *Perspectives on intellectual development: Volume 19* (pp. 131–152). Hillsdale, NJ: Lawrence Erlbaum Associates.

Goodnow, J. J. (1971). Matching auditory and visual series: Modality problem or translation problem? *Child Development, 42*, 1187–1201.

Grover, S. C. (1986). A field study of the use of cognitive-developmental principles in microcomputer design for young children. *Journal of Educational Research, 79*, 325–332.

Hargrove, P. M., & Panagos, J. M. (1982). Role saliency cues and children's sentence comprehension. *Journal of Psycholinguistic Research, 11*, 217–228.

Henriquez, A. (1988). *"Let's Learn To Play Together" testing*. Unpublished manuscript, Children's Television Workshop.

Hildebrandt, C. (1987). Structural-developmental research in music: Conservation and representation. In J. C. Peery, L. W. Peery, & T. W. Draper (Eds.), *Music and child development* (pp. 80–95). London: Springer-Verlag.

Jones, M. K. (1989). *Human-computer interaction: A design guide*. Englewood Cliffs, NJ: Educational Technology Publications.

Jones, R. L. (1976). The development of the child's conception of meter in music. *Journal of Research in Music Education, 24*(3), 142–154.

Karat, J., McDonald, J. E., & Anderson, M. (1986). A comparison of menu selection techniques: Touch panel, mouse and keyboard. *International Journal of Man-Machine Studies, 25*, 73–88.

Kingma, J., & Loth, F. L. (1984). The development of relational terms in different content areas. *Journal of General Psychology, 111*, 163–175.

Markman, E. M. (1979). Realizing that you don't understand: A preliminary investigation. *Child Development, 48*, 986–992.

Nickerson, R. S. (1986). *Using computers: Human factors in information systems*. Cambridge, MA: MIT Press.

Norman, D. A. (1988). *The psychology of everyday things*. New York: Basic Books.

Offerman, S. C. (1989). Sesame Street *First Writer testing*. Unpublished manuscript, Children's Television Workshop.

Owen, D. (1986). Naive theories of computation. In D. A. Norman & S. W. Draper (Eds.), *User centered system design* (pp. 187–200). Hillsdale, NJ: Lawrence Erlbaum Associates.

Porter, L. K., Lahm, E. A., Behrmann, M. M., & Collins, A. A. (1986). Designing picture-based software for the cognitively young. *Journal of the Division for Early Childhood, 10*(3), 231–239.

Revelle, G. L., & Strommen, E. F. (in press). The effects of practice and input device used on young children's computer control. *Journal of Computing in Childhood Education*.

Revelle, G. L., Strommen, E. F., & Offerman, S. C. (1990). *Cursor and device differences in children's computer control*. Unpublished manuscript.

Riley, M. S. (1986). User understanding. In D. A. Norman & S. W. Draper (Eds.), *User centered system design* (pp. 157–169). Hillsdale, NJ: Lawrence Erlbaum Associates.

Scardamalia, M., Bereiter, C., McLean, R. S., Swallow, J., & Woodruff, E. (1989). Computer supported intentional learning environments. *Journal of Educational Computing Research, 5*(1), 51–68.

Schauble, L. (1990). Formative evaluation in the design of educational software at the Children's Television Workshop. In B. N. Flagg (Ed.), *Formative evaluation for educational technologies* (pp. 51–66). Hillsdale, NJ: Lawrence Erlbaum Associates.

Sein, M. K., & Bostrom, R. P. (1989). Individual differences and conceptual models in training novice users. *Human-Computer Interaction, 4*, 197–229.

Shade, D. D., Nida, R. E., Lipinski, J. M., & Watson, J. A. (1986). Microcomputers and preschoolers: Working together in a classroom setting. *Computers in the Schools, 3*(2), 53–61.

Silvern, S. B., Williamson, P. A., & Countermine, T. M. (1988). Young children's interaction with a microcomputer. *Early Childhood Development and Care, 21*, 23–35.

Strommen, E. F. (1989a). *Children's performance on Big Bird's Near/Far Game*. Unpublished manuscript, Children's Television Workshop.

Strommen, E. F. (1989b). *Children's performance on The Count's I've Got Your Number Game*. Unpublished manuscript, Children's Television Workshop.

Strommen, E. F. (1989c). *Inhibition control and computer use: A research proposal*. Unpublished manuscript, Children's Television Workshop.

Strommen, E. F. (1990a). *Children's performance on Big Bird's Rhyming Game*. Unpublished manuscript, Children's Television Workshop.

Strommen, E. F. (1990b). *Children's performance on Ernie and Bert's Paperclip Chase*. Unpublished manuscript, Children's Television Workshop.

Strommen, E. F. (in progress-a). *Children's performance on the interactive compact disc, A Visit to Sesame Street, Volume I: Numbers*.

Strommen, E. F. (in progress-b). *Drawing on the television: Young children's use of an electronic drawing pad*.

Tomasello, M. (1987). Learning to use prepositions: A case study. *Journal of Child Language, 14*, 79–98.

Van Zee, N. (1976). Response of kindergarten children to musical stimuli and terminology. *Journal of Research in Music Education, 24*(1), 14–21.

View-Master Ideal. (1988). *Interactive Vision*. [VHS interactive video]. Portland, OR: Author.

Wallace, S. A., Newell, K. M., & Wade, M. G. (1978). Decision and response times as a function of movement difficulty in preschool children. *Child Development, 49*, 509–512.

Whitfield, D., Ball, R. G., & Bird, J. M. (1983). Some comparisons of on-display and off-display touch input devices for interaction with computer generated displays. *Ergonomics, 26*(11), 1033–1053.

Winner, E., Engel, M., & Gardner, H. (1980). Misunderstanding metaphor: What's the problem? *Journal of Experimental Child Psychology, 30*, 22–32.

Wright, J., Shade, D. D., Thouvenelle, S., & Davidson, J. (1989). New directions in software development for young children. *Journal of Computing in Childhood Education, 1*(1), 45–57.