26. [Introduction] Personal Dynamic Media

◊05 73

The imagination and boldness of the mid-1970s Dynabook vision—and the accuracy with which Alan Kay and Adele Goldberg foretold, in the following essay, what notebook computing has become—is striking. This introduction was composed, after all, at a café, on a computer that fits in an overcoat pocket. Before Kay and Goldberg began to outline the visions in this paper, such a possibility was seldom imagined, even in vague terms, by computing researchers. The prescience of Kay and Goldberg's vision was such that almost all the specific ideas for the uses of notebook computing developed in the group that Kay directed at Xerox Palo Alto Research Center (PARC) proved to be worthwhile. The broader idea—that the notebook computer would be a general-purpose device, with educators and businesspeople and poets all using the same type of Dynabook—has also held true.

◊16 231

The Dynabook vision came about because Kay, Goldberg, and others in their Learning Research Group at Xerox PARC considered the computer from a radically different perspective. (This approach may be the central maneuver in new media's otherwise varied methodology.) While most saw the computer as a tool for engineers or, at most, businesspeople, Kay thought computers could be used even by children, and could be used creatively. Kay and Goldberg also upset the idea that time-sharing computing is liberating for users, as J. C. R. Licklider (◊05) had more appropriately thought during the batch computing era. Instead, they believed that in the mid-1970s providing powerful, dedicated computers to individuals was a superior approach. Their group at Xerox PARC developed not only the notebook computer, but the essence of the personal desktop computer as well, which came to be embodied in Xerox's Alto. The desktop computer revolution would take hold before notebooks became widespread, of course. It was in the 1980s that home computers brought a new context to computing and revealed a host of new possible digital activities.

The development of the Dynabook vision and the powerful Alto personal computer created the elements that were used to produce the Star computer by the Xerox Systems Development Division, headed by David Liddle (formerly of PARC). The Star system sported a graphical user interface, which became part of popular personal computing via Apple's Macintosh. (Kay's fame is partially based on his invention of overlapping windows.) The Star system also helped to make Ethernet, the mouse, the laser printer, and WYIWYG printing a part of today's everyday computer environment. As with the movement of elements such as the mouse from Engelbart's ARC (◊16) to Kay's group at PARC, significant changes in focus took place in the move from the Alto to the Star. The "virtual paper" and desktop metaphors became further entrenched, while flexible knowledge spaces and user-created tools received less emphasis. Apple made changes of its own. One was in the meaning of Star's icons, which were originally only to represent documents, never applications. Apple's model was later adopted nearly wholesale by Microsoft (the exceptions were certain touches from systems such as Motif and NeXT, or from within Microsoft) and made into the dominant computing platform in the world. With today's rise in handheld computing the desktop and virtual paper metaphors are meeting a significant challenge, and may themselves fade. Still, the important original idea of opening tool creation to every user—even children—has not returned to prominence. Kay, however, continues to pursue this vision through his Squeak project.

Kay was later a research fellow at Apple and then at Disney. Before these, and after his work at PARC, he directed Atari's sizeable but short-lived research lab, which was the victim of the collapse of the U.S. video game industry in the mid-1980s.

Certain aspects of notebook computing weren't foretold in the essay that follows—even though the Dynabook idea is among the most influential and prescient of the past thirty years. While Kay and Goldberg predicted that businesspeople could carry along "the last several weeks of correspondence in a structured cross-indexed form" and wireless communications capability was an essential part of the Dynabook concept, they didn't emphasize how notebook computers (and other personal computers) would find so much use as networked communication devices. They highlighted

the potential creative uses of the computer, but did not suggest the ways in which notebook computers are now used as media players, playing DVDs in coach class on airplanes or sending sound from MP3s through earphones to students as they toil over textbooks in libraries. Even within the Dynabook project, so extraordinary in creating notebook computing and charting a course for it, the quarter-century of computer evolution that would follow was not completely prefigured.
—NM & NWF

◊09
109

Seymour Papert, a great influence on Kay, was creating computer systems for children to use creatively on the other side of the United States, at MIT. There, he developed LOGO (see ◊28). Kay's previous work on FLEX had sought to create a computer that users could program themselves. This work led to the definition of object-oriented programming (inspired, in part, by Sutherland's "Sketchpad" (◊09)). From Papert's work, Kay saw how far this idea could be carried, and refined his notion of why it was important. The next stage of Kay's work in this area culminated in Smalltalk, the environment presented in this paper. Kay wrote the following regarding Papert's influence in 1990:

"I was possessed by the analogy between print literacy and LOGO. While designing the FLEX machine I had believed that end users needed to be able to program before the computer could become truly theirs—but here was a real demonstration, and with children! The ability to 'read' a medium means you can *access* materials and tools generated by others. The ability to 'write' in a medium means you can *generate* materials and tools for others. You must have both to be literate. In print writing, the tools you generate are rhetorical; they demonstrate and convince. In computer writing, the tools you generate are processes; they simulate and decide." ("User Interface: A Personal View," 193)

◊28
413

Further Reading

Hiltzik, Michael. *Dealers of Lightning: Xerox PARC and the Dawn of the Computer Age.* New York: Harper Business, 1999.

Ingalls, Dan, Ted Kaehler, John Maloney, Scott Wallace, and Alan Kay. "Back to the Future: The Story of Squeak, A Practical Smalltalk Written in Itself." *Proceedings of the 1997 SIGPLAN Conference on Object-Oriented Programming Systems, Languages, and Applications* (OOPSLA '97), 318–326. October 1997. <ftp://st.cs.uiuc.edu/Smalltalk/Squeak/docs/OOPSLA.Squeak.html>

Kay, Alan. "User Interface: A Personal View." *The Art of Human-Computer Interface Design,* 191–207. Ed. Brenda Laurel. Reading, Mass.: Addison-Wesley, 1990.

Kay, Alan. *Doing with Images Makes Symbols.* Stanford, Calif.: University Video Communications, 1987.

Squeakland. <http://squeakland.com>

Winograd, Terry, ed. *Bringing Design to Software.* New York: Addison-Wesley, 1996.

Original Publication
Computer 10(3):31–41. March 1977.

Personal Dynamic Media

Alan Kay and Adele Goldberg

Introduction

The Learning Research Group at Xerox Palo Alto Research Center is concerned with all aspects of the communication and manipulation of knowledge. We design, build, and use dynamic media which can be used by human beings of all ages. Several years ago, we crystallized our dreams into a design idea for a personal dynamic medium the size of a notebook (the *Dynabook*) which could be owned by everyone and could have the power to handle virtually all of its owner's information-related needs. Towards this goal we have designed and built a communications system: the Smalltalk language, implemented on small computers we refer to as "interim Dynabooks." We are exploring the use of this system as a programming and problem solving tool; as an interactive memory for the storage and manipulation of data; as a text editor; and as a medium for expression through drawing, painting, animating pictures, and composing and generating music. (Figure 26.1 is a view of this interim Dynabook.)

We offer this paper as a perspective on our goals and activities during the past years. In it, we explain the Dynabook idea, and describe a variety of systems we have already written in the Smalltalk language in order to give broad images of the kinds of information-related tools that might represent the kernel of a personal computing medium.

Background

Humans and Media

"Devices" which variously store, retrieve, or manipulate information in the form of messages embedded in a medium have been in existence for thousands of years. People use them to communicate ideas and feelings both to others and back to themselves. Although thinking goes on in one's head, external media serve to materialize thoughts and, through feedback, to augment the actual paths the thinking follows. Methods discovered in one medium provide metaphors which contribute new ways to think about notions in other media.


Figure 26.1. Kids learning to use the interim Dynabook.

For most of recorded history, the interactions of humans with their media have been primarily nonconversational and passive in the sense that marks on paper, paint on walls, even "motion" pictures and television, do not change in response to the viewer's wishes. A mathematical formulation—which may symbolize the essence of an entire universe—once put down on paper, remains static and requires the reader to expand its possibilities.

Every message is, in one sense or another, a *simulation* of some idea. It may be representational or abstract. The essence of a medium is very much dependent on the way messages are embedded, changed, and viewed. Although digital computers were originally designed to do arithmetic computation, the ability to simulate the details of any descriptive model means that the computer, viewed as a medium itself, can be *all other media* if the embedding and viewing methods are sufficiently well provided. Moreover, this new "metamedium" is *active*—it can respond to queries and experiments—so that the messages may involve the learner in a two-way conversation. This property has never been available before except through the medium of an individual teacher. We think the implications are vast and compelling.

A Dynamic Medium for Creative Thought: The Dynabook

Imagine having your own self-contained knowledge manipulator in a portable package the size and shape of an ordinary notebook. Suppose it had enough power to outrace your senses of sight and hearing, enough capacity to store for later retrieval thousands of page-equivalents of reference materials, poems, letters, recipes, records, drawings, animations, musical scores, waveforms, dynamic simulations, and anything else you would like to remember and change.

We envision a device as small and portable as possible which could both take in and give out information in quantities approaching that of human sensory systems (Figure 26.2). Visual output should be, at the least, of higher quality than what can be obtained from newsprint. Audio output should adhere to similar high-fidelity standards.

There should be no discernible pause between cause and effect. One of the metaphors we used when designing such a system was that of a musical instrument, such as a flute, which is owned by its user and responds instantly and consistently to its owner's wishes. Imagine the absurdity of a one-second delay between blowing a note and hearing it!

These "civilized" desires for flexibility, resolution, and response lead to the conclusion that a user of a dynamic personal medium needs several hundred times as much power as the average adult now typically enjoys from timeshared computing. This means that we should either build a new resource several hundred times the capacity of current machines and share it (very difficult and expensive), or we should investigate the possibility of giving each person his own powerful machine. We chose the second approach.

Design Background

The first attempt at designing this metamedium (the FLEX machine[4]) occurred in 1967–69. Much of the hardware and software was successful from the standpoint of computer science state-of-the-art research, but lacked sufficient

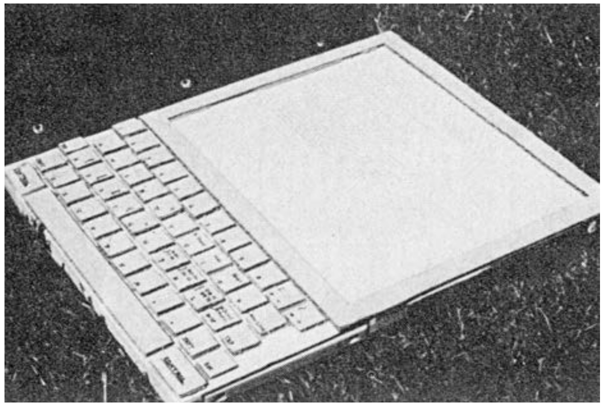


Figure 26.2. Mock-up of a future Dynabook.

expressive power to be useful to an ordinary user. At that time we became interested in focusing on children as our "user community." We were greatly encouraged by the Bolt Beranek and Newman/MIT Logo work that uses a robot turtle that draws on paper, a CRT version of the turtle, and a single music generator to get kids to program.

Considering children as the users radiates a compelling excitement when viewed from a number of different perspectives. First, the children really can write programs that do serious things. Their programs use symbols to stand for objects, contain loops and recursions, require a fair amount of visualization of alternative strategies before a tactic is chosen, and involve interactive discovery and removal of "bugs" in their ideas.

Second, the kids love it! The interactive nature of the dialogue, the fact that *they* are in control, the feeling that they are doing *real* things rather than playing with toys or working out "assigned" problems, the pictorial and auditory nature of their results, all contribute to a tremendous sense of accomplishment to their experience. Their attention spans are measured in hours rather than minutes.

Another interesting nugget was that children really needed as much or more computing power than adults were willing to settle for when using a timesharing system. The best that timesharing has to offer is slow control of crude wire-frame green-tinted graphics and square-wave musical tones. The kids, on the other hand, are used to finger-paints, water colors, color television, real musical instruments, and records. If the "medium is the message," then the message of low-bandwidth timesharing is "blah."

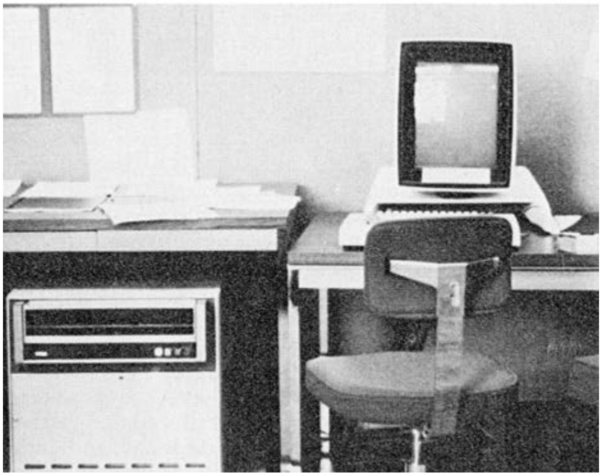


Figure 26.3. The interim Dynabook system consists of processor, disk drive, display, keyboard, and pointing devices.

An Interim Dynabook

We have designed an interim version of the Dynabook on which several interesting systems have been written in a new medium for communication, the Smalltalk programming language.[2] We have explored the usefulness of the systems with more than 200 users, most notably setting up a learning resource center in a local junior high school.

The interim Dynabook, shown in Figure 26.3, is a completely self-contained system. To the user, it appears as a small box in which a disk memory can be inserted; each disk contains about 1500 page-equivalents of manipulable storage. The box is connected to a very crisp high-resolution black and white CRT or a lower-resolution high-quality color display. Other input devices include a typewriter keyboard, a "chord" keyboard, a pointing device called a "mouse" which inputs position as it is moved about on the table, and a variety of organ-like keyboards for playing music. New input devices such as these may be easily attached, usually without building a hardware interface for them. Visual output is through the display, auditory output is obtained from a built-in digital-to-analog converter connected to a standard hi-fi amplifier and speakers.

We will attempt to show some of the kinds of things that can be done with a Dynabook; a number of systems developed by various users will be briefly illustrated. All photographs of computer output in this paper are taken from the display screen of the interim system.

Remembering, Seeing and Hearing

The Dynabook can be used as an interactive memory or file cabinet. The owner's context can be entered through a keyboard and active editor, retained and modified indefinitely, and displayed on demand in a font of publishing quality.

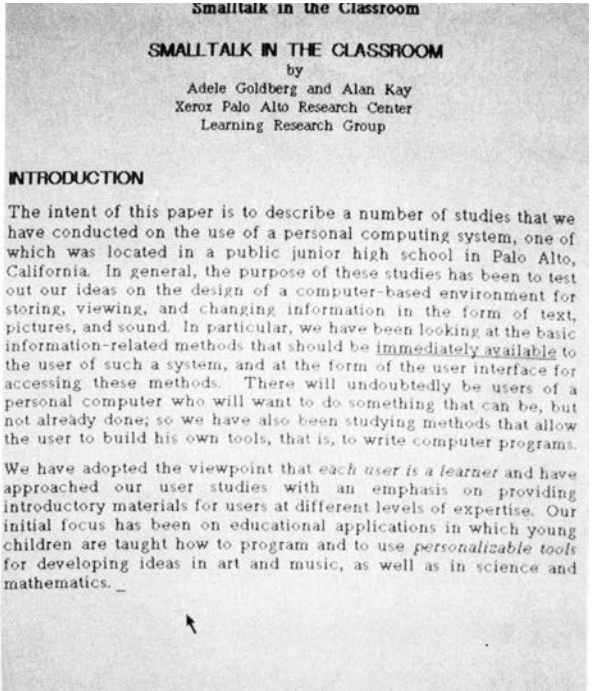
Drawing and painting can also be done using a pointing device and an iconic editor which allows easy modification of pictures. A picture is thus a manipulable object and can be animated dynamically by the Dynabook's owner.

A book can be read through the Dynabook: the memory can be inserted as shown in Figure 26.4. It need not be treated as a simulated paper book since this is a new medium with new properties. A dynamic search may be made for a particular context. The non-sequential nature of the file medium and the use of dynamic manipulation allows a story to have many accessible points of view; Durrell's *Alexandria Quartet*, for instance, could be one book in which the reader may pursue many paths through the narrative.

Figure 26.4. Inserting the disk pack in the Dynabook.

Different Fonts for Different Effects

One of the goals of the Dynabook's design is *not* to be *worse* than paper in any important way. Computer displays of the past have been superior in matters of dynamic writing and erasure, but have failed in contrast, resolution, or ease of viewing. There is more to the problem than just the display of text in a high-quality font. Different fonts create different moods and cast an aura that influences the subjective style of both writing and reading. The Dynabook is supplied with a number of fonts which are contained on the file storage. Figures 26.5 and 26.6 show samples of these fonts.



Smalltalk in the Classroom

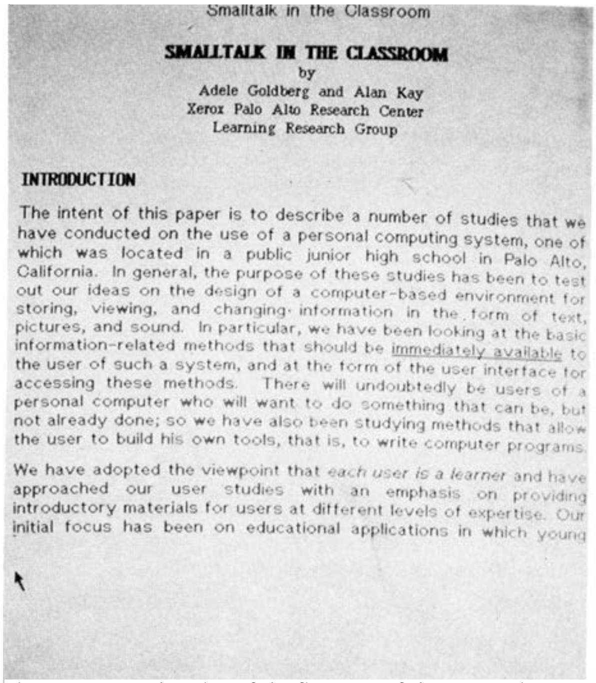
SMALLTALK IN THE CLASSROOM
by
Adele Goldberg and Alan Kay
Xerox Palo Alto Research Center
Learning Research Group

INTRODUCTION

The intent of this paper is to describe a number of studies that we have conducted on the use of a personal computing system, one of which was located in a public junior high school in Palo Alto, California. In general, the purpose of these studies has been to test out our ideas on the design of a computer-based environment for storing, viewing, and changing information in the form of text, pictures, and sound. In particular, we have been looking at the basic information-related methods that should be immediately available to the user of such a system, and at the form of the user interface for accessing these methods. There will undoubtedly be users of a personal computer who will want to do something that can be, but not already done; so we have also been studying methods that allow the user to build his own tools, that is, to write computer programs.

We have adopted the viewpoint that *each user is a learner* and have approached our user studies with an emphasis on providing introductory materials for users at different levels of expertise. Our initial focus has been on educational applications in which young children are taught how to program and to use *personalizable tools* for developing ideas in art and music, as well as in science and mathematics. _

Figure 26.5. First page of this paper as photographed from the display screen.



Smalltalk in the Classroom

SMALLTALK IN THE CLASSROOM
by
Adele Goldberg and Alan Kay
Xerox Palo Alto Research Center
Learning Research Group

INTRODUCTION

The intent of this paper is to describe a number of studies that we have conducted on the use of a personal computing system, one of which was located in a public junior high school in Palo Alto, California. In general, the purpose of these studies has been to test out our ideas on the design of a computer-based environment for storing, viewing, and changing information in the form of text, pictures, and sound. In particular, we have been looking at the basic information-related methods that should be immediately available to the user of such a system, and at the form of the user interface for accessing these methods. There will undoubtedly be users of a personal computer who will want to do something that can be, but not already done; so we have also been studying methods that allow the user to build his own tools, that is, to write computer programs.

We have adopted the viewpoint that *each user is a learner* and have approached our user studies with an emphasis on providing introductory materials for users at different levels of expertise. Our initial focus has been on educational applications in which young

Figure 26.6. Another view of the first page of the paper using different fonts.

The Dynabook as a personal medium is flexible to the point of allowing an owner to choose his own ways to view information. Any character font can be described as a matrix of black and white dots. The owner can draw in a character font of his own choosing. He can then immediately view font changes within the context of text displayed in a window. With the Dynabook's fine grain of display, the rough edges disappear at normal viewing distance to produce high-quality characters.

The malleability of this approach is illustrated in Figure 26.7: this owner has decided to embellish some favorite nouns with their iconic referent. Such a facility would be useful in enhancing an early reading curriculum.

Editing

Every description or object in the Dynabook can be displayed and edited. Text, both sequential and structured, can easily be manipulated by combining pointing and a simple "menu" for commands, thus allowing deletion, transposition, and structuring. Multiple windows, as shown in Figure 26.8, allow a document (composed of text, pictures, musical notation) to be created and viewed simultaneously at several levels of refinement. Editing operations on other viewable objects (such as pictures and fonts) are handled in analogous ways.

Filing

The multiple-window display capability of Smalltalk has inspired the notion of a dynamic *document.* A document is a collection of objects that have a sensory display and have something to do with each other; it is a way to store and retrieve related information. Each subpart of the document, or *frame,* has its own editor which is automatically invoked when pointed at by the "mouse." These frames may be related sequentially, as with ordinary paper usage, or *inverted* with respect to properties, as in cross-indexed file systems. *Sets* which can automatically map their contents to secondary storage with the ability to form unions, negations, and intersections are part of this system, as is a "modeless" text editor with automatic right justification.

The current version of the system is able to automatically cross-file several thousand multifield records (with formats chosen by the user), which include ordinary textual documents indexed by content, the Smalltalk system, personal files, diagrams, and so on. (See Figures 26.9–12.)

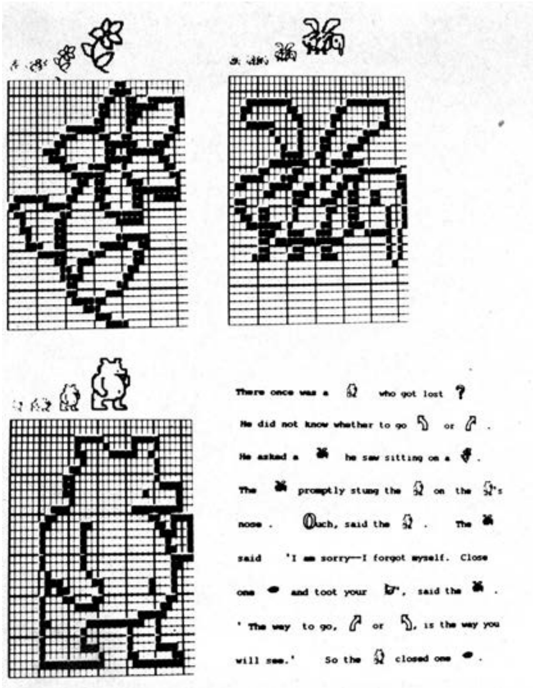


Figure 26.7. Fonts for a bear, a flower, and a bee used to tell a story with pictures.

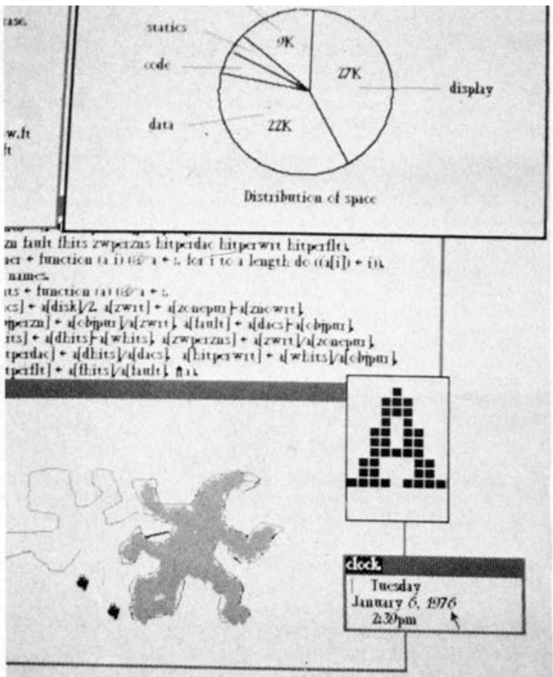


Figure 26.8. Multiple windows allow documents containing text and pictures to be created and viewed.

Drawing/Painting

The many small dots required to display high-quality characters (about 500,000 for an 8-½″ × 11″ sized display) also allow sketching-quality drawing, "halftone painting," and animation. The subjective effect of gray scale is caused by the eye fusing an area containing a mixture of small black and white dots. The pictures in Figures 26.13 and 26.14 show a palette of toned patterns with some brushes. A brush can be grabbed with the "mouse," dipped into a paint pot, and then the halftone can be swabbed on as a function of the size, shape, and velocity of the brush. The last pair of pictures shows a heart/peace symbol shaped brush used to give the effect of painting wallpaper.

Curves are drawn by a *pen* on the display screen. (Straight lines are curves with zero curvature.) In the Dynabook, *pens* are members of a class that can selectively draw with black or white (or colored) ink and change the thickness of the trace. Each *pen* lives in its own *window,* careful not to traverse its window boundaries but to adjust as its window changes size and position. This window idea is illustrated in Figure 26.15; a number of simple and elegant examples are displayed in the windows.

Animation and Music

Animation, music, and programming can be thought of as different *sensory views* of dynamic processes. The structural similarities among them are apparent in Smalltalk, which provides a common framework for expressing those ideas.

All of the systems are equally controllable by hand or by program. Thus, drawing and painting can be done using a pointing device or in conjunction with programs which draw curves, fill in areas with tone, show perspectives of three-dimensional models (see Figure 26.16), and so on. Any graphic expression can be animated, either by reflecting a simulation or by example (giving an "animator" program a sample trace or a route to follow).

Music is controlled in a completely analogous manner. The Dynabook can act as a "super synthesizer" getting direction either from a keyboard or from a "score." The keystrokes can be captured, edited, and played back. Timbres, the "fonts" of musical expression, contain the quality and mood which different instruments bring to an orchestration. They may be captured, edited, and used dynamically.

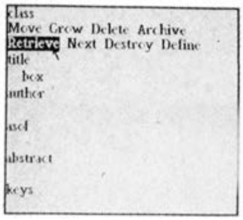


Figure 26.9. Retrieval in this filing tool is carried out by pointing to the command in the documents menu. The system will find every document with the title "box."

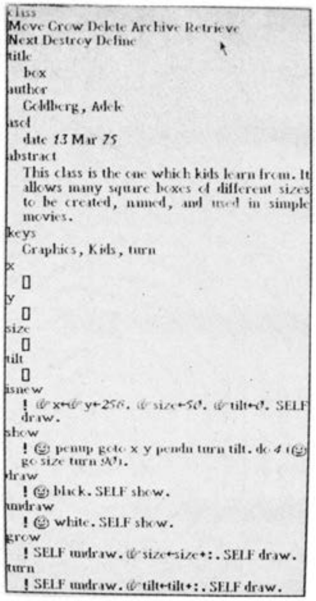


Figure 26.10. Here is a retrieved document that represents a description of a Smalltalk class definition.

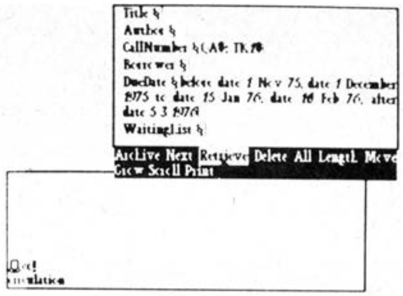


Figure 26.12. This retrieval request combines incomplete call numbers with date ranges. The example is taken from an experimental library system.

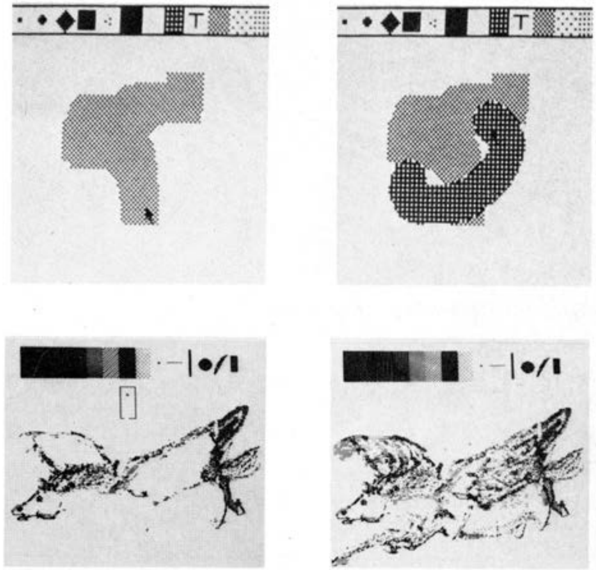


Figure 26.13. A sketch of Pegasus is shown being drawn with a Smalltalk drawing tool. The first two pictures in the sequence show halftone "paint" being scrubbed on.

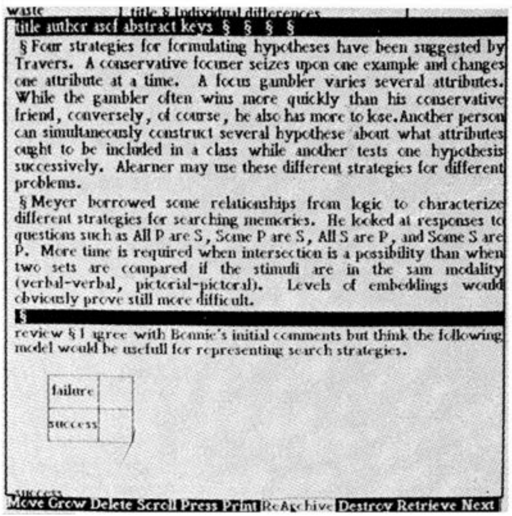


Figure 26.11. This is a document from an annotated bibliography for teacers. Details are suppressed but can be expanded by pointing to names in the black fields. Documents can also contain diagrams.

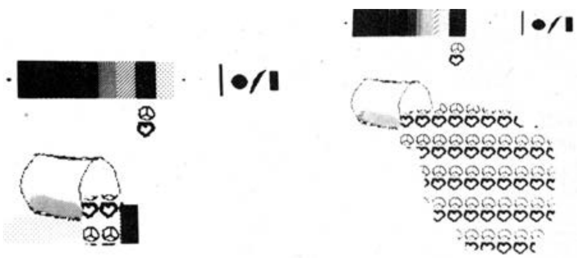


Figure 26.14. A sketch of a heart/peace symbol is created and used as a paint brush.

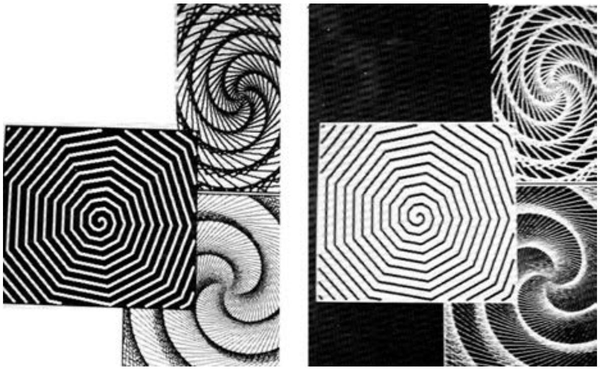


Figure 26.15. Curves can be drawn using Smalltalk line-drawing commands. These curves are constrained to show in display windows. Black and white can be reversed for interesting effects.

Simulation

In a very real sense, simulation is the central notion of the Dynabook. Each of the previous examples has shown a simulation of visual or auditory media. Here are a number of examples of interesting simulations done by a variety of users.

An Animation System Programmed by Animators

Several professional animators wanted to be able to draw and paint pictures which could then be animated in real time by simply showing the system roughly what was wanted.

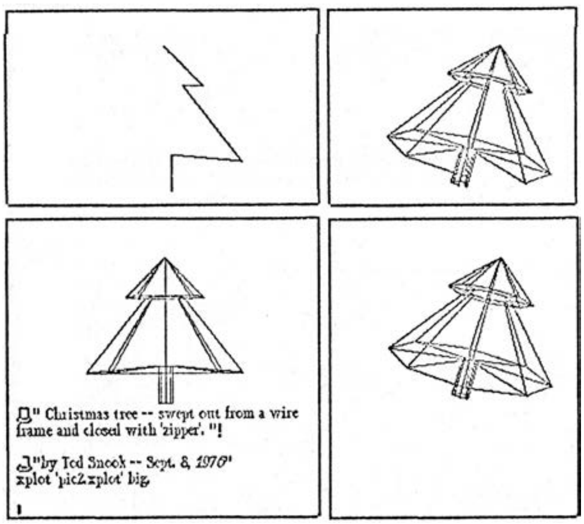


Figure 26.16. A model of three-dimensional graphics as implemented in Smalltalk.

Desired changes would be made by iconically editing the animation sequences.

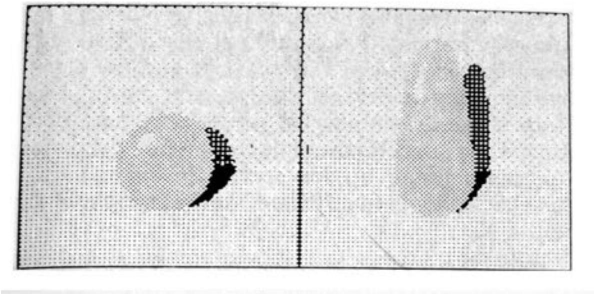
Much of the design of SHAZAM, their animation tool, is an automation of the media with which animators are familiar: *movies* consisting of sequences of *frames* which are a composition of transparent *cels* containing foreground and background drawings. Besides retaining these basic concepts of conventional animation, SHAZAM incorporates some creative supplementary capabilities.

Animators know that the main action of animation is due not to an individual frame, but to the change from one frame to the next. It is therefore much easier to plan an animation if it can be seen moving as it is being created. SHAZAM allows any cel of any frame in an animation to be edited while the animation is in progress. A library of already-created cels is maintained. The animation can be single-stepped; individual cels can be repositioned, reframed, and redrawn; new frames can be inserted; and a frame sequence can be created at any time by attaching the cel to the pointing device, then *showing* the system what kind of movement is desired. The cels can be stacked for background parallax; *holes* and *windows* are made with *transparent* paint. Animation objects can be painted by programs as well as by hand. The control of the animation can also be easily done from a Smalltalk simulation. For example, an animation of objects bouncing in a room is most easily accomplished by a few lines of Smalltalk that express the class of bouncing objects in physical terms. Figures 26.17, 18, and 19 show some animations created by young children.

A Drawing and Painting System Programmed by a Child

One young girl, who had never programmed before, decided that a pointing device *ought* to let her draw on the screen. She then built a sketching tool without ever seeing ours (displayed in Figure 26.20). She constantly embellished it with new features including a menu for brushes selected by pointing. She later wrote a program for building tangram designs (Figure 26.21).

This girl has taught her own Smalltalk class; her students were seventh-graders from her junior high school. One of them designed an even more elaborate system in which pictures are constructed out of geometric shapes created by pointing to a menu of commands for creating regular



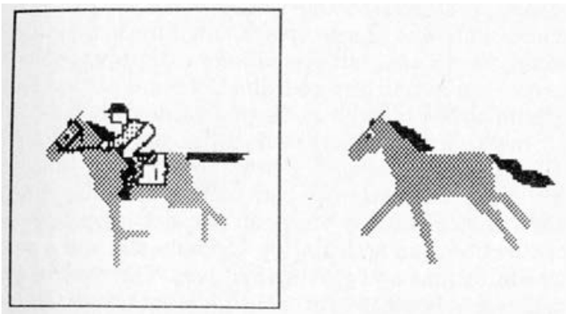
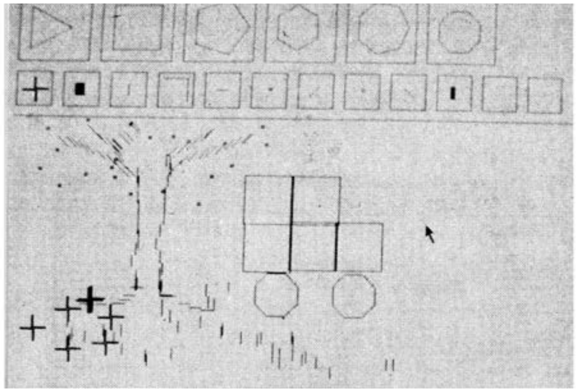


Figure 26.17. An animation of a drop of water.

Figure 26.18. An animation of a galloping horse, with and without a rider.

Figure 26.19. An animation of a frog catching a fly.



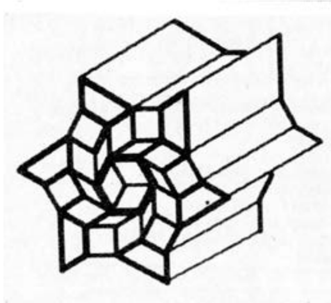


Figure 26.20. One of the first painting tools designed and implemented in Smalltalk by a twelve-year-old girl.

Figure 26.21. Tangram designs are created by selecting shapes from a "menu" displayed at the top of the screen. This system was implemented in Smalltalk by a fourteen-year-old girl.

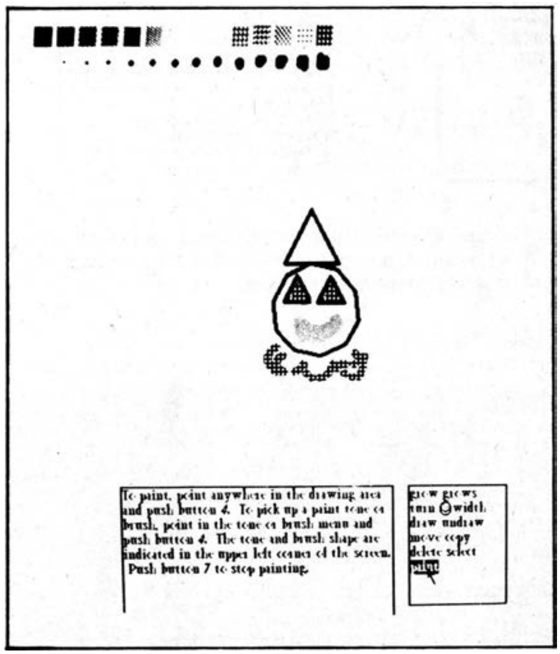


Figure 26.22. In this young student's Smalltalk painting system, pictures are constructed out of geometric shapes.

polygons (Figure 26.22). The polygons can then be relocated, scaled, and copied; their color and line width can change.

A Hospital Simulation Programmed by a Decision-Theorist

The simulation shown in Figure 26.23 represents a hospital in which every *department* has resources which are used by *patients* for some *duration of time.* Each patient has a *schedule* of departments to visit; if there are no resources available (doctors, beds), the patient must *wait* in line for service. The Smalltalk description of this situation involves the class of *patients* and the class of *departments.* The generalization to any hospital configuration with any number of patients is part of the simulation. The particular example captured in the pictures shows patients lining up for service in *emergency*. It indicates that there is insufficient staff available in that important area.

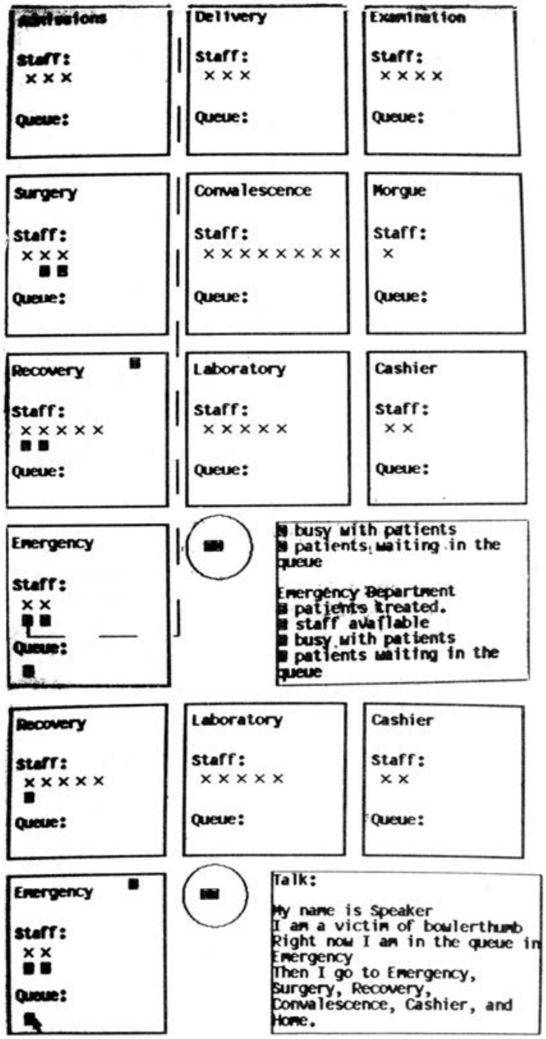


Figure 26.23. A view of a simulation of a hospital. The rectangles represent departments in which staff members (X's) treat patients (numbers).

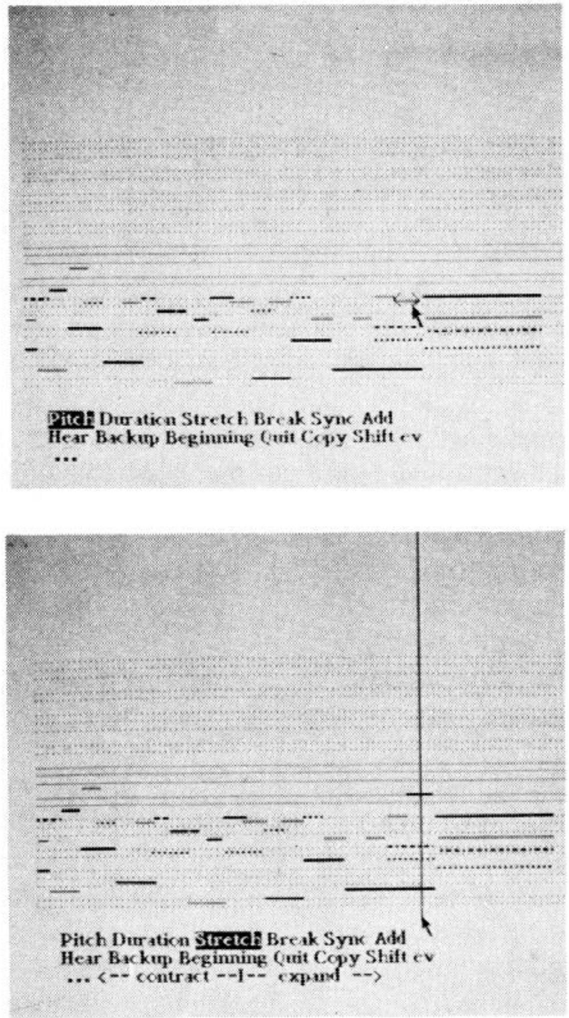


Figure 26.24. These two pictures show a musical score being edited. A note is selected in order to change its pitch. Next, the score is stretched, that is, the notes at the selected position will be held for a longer duration.

An Audio Animation System Programmed by Musicians

Animation can be considered to be the coordinated parallel control through time of images conceived by an animator. Likewise, a system for representing and controlling musical images can be imagined which has very strong analogies to the visual world. Music is the design and control of images (pitch and duration changes) which can be *painted* different *colors* (timbre choices); it has synchronization and coordination, and a very close relationship between audio and spatial visualization.

The Smalltalk model created by several musicians, called TWANG, has the notion of a *chorus* which contains the main control directions for an overall piece. A chorus is a kind of *rug* with a warp of parallel sequences of "pitch, duration, and articulation" commands, and a woof of synchronizations and global directives. The control and the *player* are separate: in SHAZAM, a given movie sequence can animate many

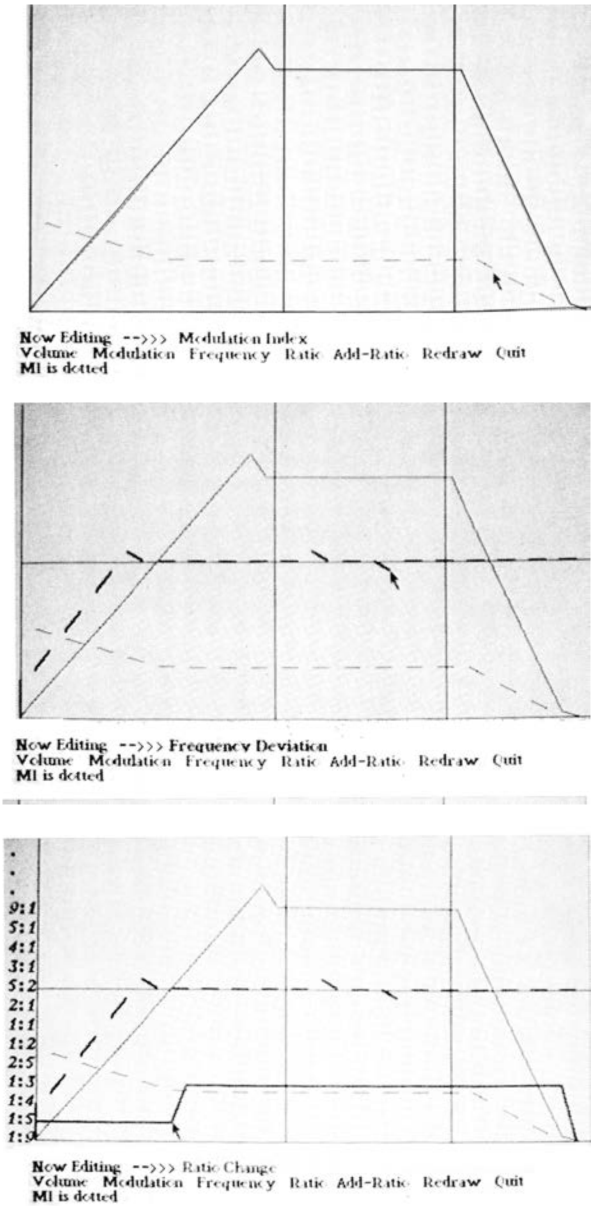


Figure 26.25. Timbre editing: a musical instrument is created by specifying the frequency, amplitude, and spectrum of its sound during a period of a few seconds. The solid line in the first picture represents volume. The first segment of the graph represents the initial attack of each note, the part between vertical bars will be repeated as long as the note is held, and the remainder will be heard as the decay.

drawings; in TWANG, a given chorus can tell many different kinds of instrumentalists what should be played. These *voices* can be synthetic timbres or timbres captured from real instruments. Musical effects such as vibrato, portamento, and diminuation are also available.

A chorus can be *drawn* using the pointing device, or it can be *captured* by playing it on a keyboard. It can be played back in real time and dynamically edited in a manner very similar to the animation system. The accompanying set of pictures in Figure 26.24 are excerpts from a sequence in which a user plays, edits, and replays a piece.

We use two methods for real-time production of high-quality timbres; both allow arbitrary transients and many independent parallel voices, and are completely produced by programs. One of these allows independent dynamic control of the spectrum, the frequency, the amplitude, and the particular collection of partials which will be heard (illustrated in Figure 26.25).

For children, this facility has a number of benefits: the strong similarities between the audio and visual worlds are emphasized because a single vernacular *which actually works* in both worlds is used for description; and second, the arts and skills of composing can be learned at the same time since tunes may be drawn in by hand and played by the system. A line of music may be copied, stretched, and shifted in time and pitch; individual notes may be edited. Imitative counterpoint is thus easily created by the fledgling composer.

A Musical Score Capture System Programmed by a Musician

OPUS is a musical score capture system that produces a display of a conventional musical score from data obtained by playing a musical keyboard. OPUS is designed to allow incremental input of an arbitrarily complicated score (full orchestra with chorus, for example), editing pages of the score, and hard copy of the final result with separate parts for individual instruments. The picture in Figure 26.26 shows a score captured with the OPUS system.

Electronic Circuit Design by a High School Student

Using several kinds of iconic menus, this student system lets the user lay out a sophisticated electronic circuit, complete with labels (Figure 26.27).

Figure 26.26. Data for this score was captured by playing on a musical keyboard. A program then converts the data to standard musical notation.

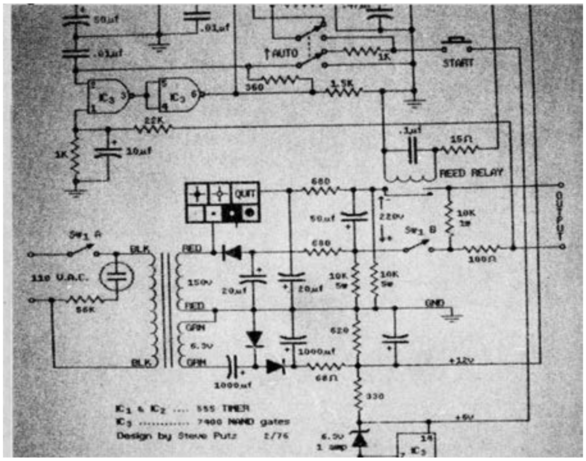


Figure 26.27. An electronic circuit layout system programmed by a 15-year-old student. Circuit components selected from a menu are moved into position with the mouse, connected to other components with lines, and labeled with text. The rectangle in the center of the layout is an iconic menu from which line widths and connectors (solid and open) may be chosen.

Conclusion

What would happen in a world in which everyone had a Dynabook? If such a machine were designed in a way that *any* owner could mold and channel its power to his own needs, then a new kind of medium would have been created: a metamedium, whose content would be a wide range of already-existing and not-yet-invented media.

An architect might wish to simulate three-dimensional space in order to peruse and edit his current designs, which could be conveniently stored and cross-referenced.

A doctor could have on file all of his patients, his business records, a drug reaction system, and so on, all of which could travel with him wherever he went.

A composer could hear his composition while it was in progress, particularly if it were more complex than he was able to play. He could also bypass the incredibly tedious chore of redoing the score and producing the parts by hand.

Learning to play music could be aided by being able to capture and hear one's own attempts and compare them against expert renditions. The ability to express music in visual terms which could be filed and played means that the acts of composition and self-evaluation could be learned without having to wait for technical skill in playing.

Home records, accounts, budgets, recipes, reminders, and so forth, could be easily captured and manipulated.

Those in business could have an active briefcase which travelled with them, containing a working simulation of their company, the last several weeks of correspondence in a structured cross-indexed form—a way to instantly calculate profiles for their futures and help make decisions.

For educators, the Dynabook could be a new world limited only by their imagination and ingenuity. They could use it to show complex historical inter-relationships in ways not possible with static linear books. Mathematics could become a living language in which children could cause exciting things to happen. Laboratory experiments and simulations too expensive or difficult to prepare could easily be demonstrated. The production of stylish prose and poetry could be greatly aided by being able to easily edit and file one's own compositions.

These are just a few ways in which we envision using a Dynabook. But if the projected audience is to be "everyone," is it possible to make the Dynabook generally useful, or will it collapse under the weight of trying to be too many different tools for too many people? The total range of possible users is so great that any attempt to specifically anticipate their needs in the design of the Dynabook would end in a disastrous feature-laden hodgepodge which would not be really suitable for anyone.

Some mass items, such as cars and television sets, attempt to anticipate and provide for a variety of applications in a fairly inflexible way; those who wish to do something different will have to put in considerable effort. Other items, such as paper and clay, offer many dimensions of possibility and high resolution; these can be used in an unanticipated way by many, though *tools* need to be made or obtained to stir some of the medium's possibilities while constraining others.

We would like the Dynabook to have the flexibility and generality of this second kind of item, combined with tools which have the power of the first kind. Thus a great deal of effort has been put into providing both endless possibilities and easy tool-making through the Smalltalk programming language.

Our design strategy, then, divides the problem. The burden of system design and specification is transferred to the user. This approach will only work if we do a very careful and comprehensive job of providing a general medium of communication which will allow ordinary users to casually and easily describe their desires for a specific tool. We must also provide enough already-written general tools so that a user need not start from scratch for most things she or he may wish to do.

We have stated several specific goals. In summary, they are:

- to provide coherent, powerful examples of the use of the Dynabook in and across subject areas;
- to study how the Dynabook can be used to help expand a person's visual and auditory skills;
- provide exceptional freedom of access so kids can spend a lot of time probing for details, searching for a personal key to understanding processes they use daily; and
- to study the unanticipated use of the Dynabook and Smalltalk by children in all age groups.

Note

An expanded version of this paper was produced as Xerox PARC Technical Report SSL-76-1, March, 1976.[5]

References and Bibliography

The following is a list of references that provides further details on some of the different systems described in this report.

1. Baeker, Ronald, "A Conversational Extensible System for the Animation of Shaded Images," *Proc. ACM SIGGRAPH Symposium,* Philadelphia, Pennsylvania, June 1976.

2. Goldberg, Adele and Alan Kay (Eds.), *Smalltalk-72 Instruction Manual,* Xerox Palo Alto Research Center, Technical Report No. SSL 76-6, March 1976.

3. Goldeen, Marian, "Learning About Smalltalk," *Creative Computing,* September–October 1975.

4. Kay, Alan, *The Reactive Engine,* Doctoral dissertation, University of Utah, September 1969.

5. Learning Research Group, "Personal Dynamic Media," Xerox Palo Alto Research Center, Technical Report No. SSL 76-1, March 1976.

6. Saunders, S., "Improved FM Audio Synthesis Methods for Realtime Digital Music Generation," *Proc., ACM Computer Science Conference,* Washington, D.C., February 1975.

7. Smith, David C., *PYGMALION: A Creative Programming Environment*, Doctoral dissertation, Stanford University Computer Science Department, June 1975.

8. Snook, Tod, *Three-dimensional Geometric Modelling,* Masters thesis, University of California, Berkeley, September 1976.