The Archi tec ture Ma chine

Nicholas Negroponte

The Architecture Machine

The Architecture Machine

Toward A More Human Environment

Nicholas Negroponte

The MIT Press
Cambridge,
Massachusetts, and
London,
England

Printed and bound
in the United States
of America.

Open access edition funded by the National Endowment for the Humanities/Andrew W. Mellon Foundation Humanities Open Book Program.

ISBN 0 262 14008 X (hardcover)
ISBN 0 262 64010 4 (paperback)

Library of Congress catalog
card number: 75-95283

To the first machine that can appreciate the gesture.

A Preface to a Preface

You will find that this book is all beginning and no end.

Most of the machines I will be discussing do not exist at this time. The chapters are primarily extrapolations into the future derived from experiences with various computer-aided design systems and, in particular, URBAN5. Some of the bents and biases may suffer from provincialism in that they reflect a general unhappiness on my part with the present practice of architecture.

There are three possible ways in which machines can assist the design process: (1) current procedures can be automated, thus speeding up and reducing the cost of existing practices; (2) existing methods can be altered to fit within the specifications and constitution of a machine, where only those issues are considered that are supposedly machine-compatible; (3) the design process, considered as evolutionary, can be presented to a machine, also considered as evolutionary, and a mutual training, resilience, and growth can be developed.

I shall consider only the third alternative and shall treat the problem as the intimate association of two dissimilar species (man and machine), two dissimilar processes (design and computation), and two intelligent systems (the architect and the architecture machine). By virtue of ascribing intelligence to an artifact or the artificial, the partnership is not one of master and slave but rather of two associates that have a potential and a desire for self-improvement.

Given that the physical environment is not in perfect harmony with every man's life style, given that architecture is not the faultless response to human needs, given that the architect is not the consummate manager of physical environments, I shall consider the physical environment as an evolving organism as opposed to a designed artifact. In particular, I shall consider an evolution aided by a specific class of machines. Warren McCulloch (1956) calls them ethical robots; in the context of architecture I shall call them architecture machines.

The Architecture Machine is for students, for people who are interested in groping with problems they do not know how to handle and asking questions they do not know how to answer. Those people who know how computers should be used in architecture, or those who expect to find the answers in this volume, should not read on. This work results from playing and fumbling with both good and bad ideas. It is not a definitive work or magnum opus on the subject of computer-aided architecture or robot architects.

Nicholas Negroponte, May 1969

Acknowledgments

Much of teaching today is no longer the presentation by one who has the word to many who do not. Teaching is a joint searching; there can be no distinction between course work and project work, research and teaching. They are inseparable, and their contributions to this book are inseparable. Therefore many people who have contributed to this book will remain anonymous, because there are indeed so many. Most of them are students.

The work that led to this book has been conducted under the joint sponsorship of Dean Lawrence Anderson (School of Architecture and Planning, M.I.T.), Dean Gordon Brown (School of Engineering, M.I.T.), and Mr. Norman Rasmussen (I.B.M. Cambridge Scientific Center). During the writing of the manuscript I have received generous support from the Harvard and M.I.T. Joint Center for Urban Studies and the M.I.T. Urban Systems Laboratory.

Dr. Warren Brodey, Professor Seymour Papert, and Professor Steven Coons have provided the theoretical foundations for many of the concepts contained in this book. In addition, Dr. Oliver Selfridge, Dr. Avery Johnson, Professor William Porter, Mr. Stuart Silverstone, Mr. Timothy Johnson, and Mr. Craig Johnson have generously participated. Professor Donlyn Lyndon, Professor Aaron Fleisher, and Professor Imre Halasz deserve many thanks for their combined patience and severity in the early days of the manuscript. They especially helped with the soul-searching task of articulating the future in the present tense.

Finally the reader should know that the entire contents of this book are not uniquely my own. Professor Leon Groisser is coauthor of almost every idea and has been my partner in this venture for five years. His formal participation in composing the text has been hampered only by a concurrent commitment to another dissertation.

N.N.

The Architecture Machine

000

Introduction

Humanism through Intelligent Machines

. . . so much corn, so much cloth, so much everything, that things will be practically without price. There will be no poverty. All work will be done by living machines. Everybody will be free from worry and liberated from the degradation of labor. Everybody will live only to perfect himself.
Karel Capek, *Rossum's Universal Robots*

Computer-aided design cannot occur without machine intelligence—and would be dangerous without it. In our era, however, most people have serious misgivings about the feasibility and more importantly, the desirability of attributing the actions of a machine to intelligent behavior. These people generally distrust the concept of machines that approach (and thus why not pass?) our own human intelligence. In our culture an intelligent machine is immediately assumed to be a bad machine. As soon as intelligence is ascribed to the artificial, some people believe that the artifact will become evil and strip us of our humanistic values. Or, like the great gazelle and the water buffalo, we will be placed on reserves to be pampered by a ruling class of automata.

Why ask a machine to learn, to understand, to associate courses with goals, to be self-improving, to be ethical—in short, to be intelligent?

The answer is the underlying postulate of an architecture machine. A design machine must have an artificial intelligence because any design procedure, set of rules, or truism is tenuous, if not subversive, when used out of context or regardless of context. It follows that a mechanism must recognize and understand the context before carrying out an operation. Therefore, a machine must be able to discern changes in meaning brought about by changes in context, hence, be intelligent (A. Johnson, 1969c). And to do this, it must have a sophisticated set of sensors, effectors, and processors to view the real world directly and indirectly.

Intelligence is a behavior. It implies the capacity to add to, delete from, and use stored information. What makes this behavior unique and particularly difficult to emulate in machines is its extreme dependence on context: time, locality, culture, mood, and so forth. For example, the meaning of a literary metaphor is conveyed through context; assessment of such meaning is an intelligent act. A metaphor in a novel characterizes the time and culture in which it was written.

One test for machine intelligence, though not necessarily machine maturity, wisdom, or knowledge, is the machine's ability to appreciate a joke. The punch line of a joke is an about-face in context; as humans we exhibit an intelligence by tracing back through the previous metaphors, and we derive pleasure from the new and surprising meanings brought on by the shift in context. People of different cultures have difficulty understanding each other's jokes.

Some architects might propose that machines cannot design unless they can think, cannot think unless they want, and cannot want unless they have bodies; and, since they do not have bodies, they therefore cannot want; thus cannot

The Spanish colonials laid out entire cities with enough megalomania to accommodate expansion for many centuries. These cities were usually designed by small bands of soldiers whose design skill was limited to a book of rules. Accordingly, irrespective of context, giant grids were decreed as a result of "global goals" such as riot control and religious prominence.

The two illustrations are of LaPaz, Bolivia. The top photograph shows the central city, which still conforms to the original scheme. The bottom photograph shows expansion to the north. It is interesting to note that this growth beyond the Spanish colonial plan has forced a "pebble-oriented" architecture. This is caused by two shifts in context: one of time and one of terrain.

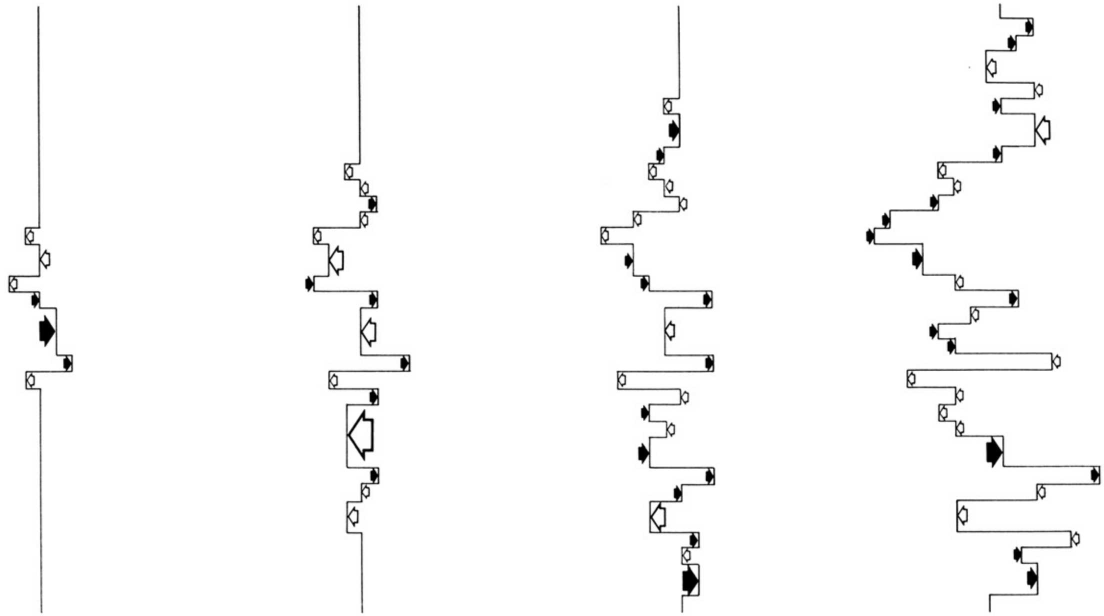
think, thus cannot design: *quod erat demonstrandum.* This argument, however, is usually emotional rather than logical. Nonetheless, the reader must recognize, if he is an "artificial intelligence" enthusiast, that intelligent machines do not exist today and that theories of machine intelligence at this time can at best be substantiated with such an example as a computer playing a superb game of checkers (Samuel, 1967) and a good game of chess (Greenblatt, et al., 1967). Furthermore, architecture, unlike a game of checkers with fixed rules and a fixed number of pieces, and much like a joke, determined by context, is the croquet game in *Alice in Wonderland,* where the Queen of Hearts (society, technology, economics) keeps changing the rules.

In the past when only humans were involved in the design process, the absence of resolute rules was not critical. Being an adaptable species, we have been able to treat each problem as a new situation, a new context. But machines at this point in time are not very adaptable and are prone to encourage repetition in process and repetition in product. The result is often embodied in a simple procedure that is computerized, used over and over, and then proves to be immaterial, irrelevant, and undesirable.

Ironically, though it is now difficult for a machine to have adaptable methods, machines can be employed in a manner that treats pieces of information individually and in detail. Imagine a machine that can respond to local situations (a family that moves, a residence that is expanded, an income that decreases). It could report on and concern itself specifically with the unique and the exceptional. It would concentrate on the particulars, "for particulars, as everyone knows, make for virtue and happiness; generalities are intellectually necessary evils" (Huxley, 1939). Human designers cannot do this; they cannot accommodate the particular, instead they accommodate the general. "He (the architect) is forced to proceed in this way because the effectuation of planning requires rules of general applicability and because watching each sparrow is too troublesome for any but God" (Harris, 1967a).

Consider a beach formed of millions of pebbles; each has a specific color, shape, and texture. A discrete pebble could have characteristics, for example, black, sharp, hard. At the same time the beach might be generally described as beige, rolling, soft. Humans learn particulars and remember generalities, study the specific and act on the general, and in this case the general conflicts with the particular. The problem is therefore twofold: first, architects cannot handle large-scale problems (the beaches) for they are too complex; second, architects ignore small-scale problems (the pebbles) for they are too particular and individual. Architects do not appear to be well trained to look at the whole urban scene; nor are they apparently skilled at observing the needs of the particular, the family, the individual. As a result "less than 5 percent of the housing built in the United States and less than 1 percent of the urban environment is exposed to the skills of the design professions" (Eberhard, 1968b).

But architects do handle "building-size" problems, a kind of concern that too often competes with general goals and at the same time couches



1

2

3

4

personal needs in antihuman structures. The result is an urban monumentalism that, through default, we have had foisted upon us by opulent, self-important institutions (that can at least control large chunks of the beach); our period is a period of neo-Hancockism and post-Prudentialism. The cause is the distinct maneuverability gap that exists between the scale of the mass and the scale of the individual, the scale of the city and the scale of the room.

Because of this, an environmental humanism might only be attainable in cooperation with machines that have been thought to be inhuman devices but in fact are devices that can respond intelligently to the tiny, individual, constantly changing bits of information that reflect the identity of each urbanite as well as the coherence of the city. These devices need the adaptability of humans and the specificity of present-day machines. They must recognize general shifts in context as well as particular changes in need and desire.

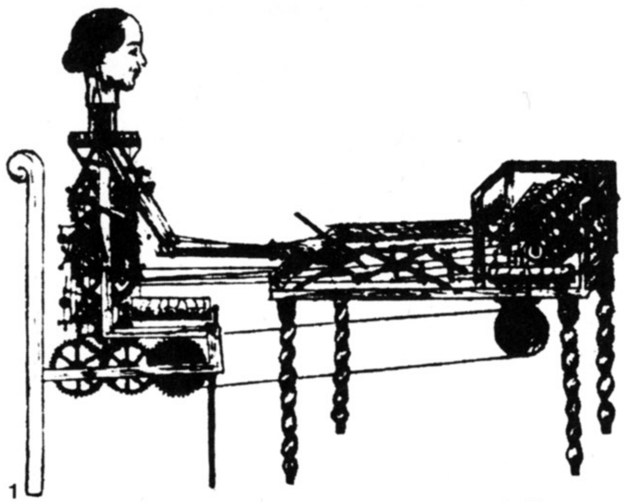
The following chapters have a "pebble-prejudice." Most computer-oriented tasks today are the opposite: the efficient transportation system, the public open space, the flow of goods and money. Our bias toward localized information implies two directions for the proposed relationship between designer and machine. The first is a "do-it-yourselfism," where, as in the Marshall McLuhan (1965) automation circuit, consumer becomes producer and dweller becomes designer. Machines located in homes could permit each resident to project and overlay his architectural needs upon the changing framework of the city. The same machine might report the number of shopping days

1 The diagram is a metaphor. The many little forces are not summed or averaged, rather they are constantly and individually affecting a single body. It is this multitude of forces, causes, and effects that the machine can so readily handle as individual events in a particular context.

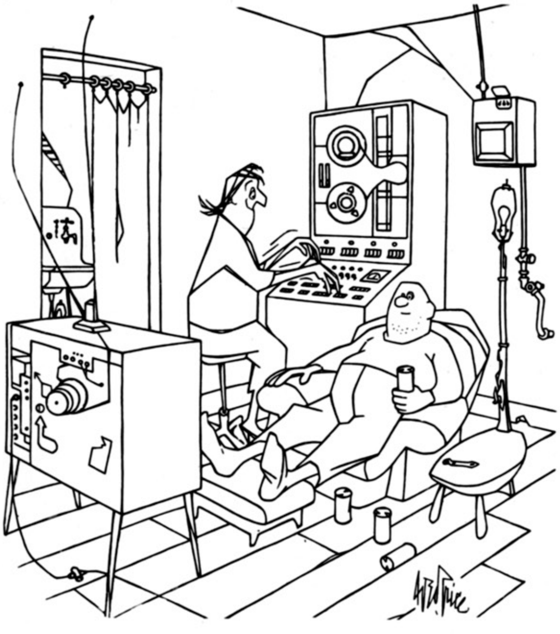
2 Handling design problems solely at the building scale can provide a monumentalism by ignoring all the local forces. Of course, Brasilia works, but only as a symbolic statement of power and not as a place to live and work. It is the result of global and general (and perhaps unethical) goals housed at the wrong scale.

3 Mojacar in the province of Almeria, Spain. This is an example of local forces shaping the environment. The unity, which results from more global causes, comes from the limitation of materials, resources, weather, and so on. (The photograph first appeared in *Architecture without Architects* [Rudofsky, 1964]. Photograph courtesy of José Ortiz Echagüe)

4 Italian hill towns. "The very thought that modern man could live in anachronistic communities like these [Positano, Italy] would seem absurd were it not that they are increasingly becoming refuges for city dwellers" (Rudofsky, 1964). The unmentioned amenities are in fact attainable in high-density urban life, now that the serial, repititious, and generalized aspects of the industrial revolution can be superseded. (Photograph courtesy of Gabinetto Fotografico Nazionale, Rome, Italy)



1



2

1 Trick automaton feigning to write, draw, and calculate, made by Leon Joly (circa 1855). (Illustration courtesy of Editions du Griffon, Neuchâtel, Switzerland)

2 The computer at home is not a fanciful concept. As the cost of computation lowers, the computer utility will become a consumer item, and every child should have one. (Cartoon from January 13, 1968, issue of Business Week. Courtesy of George Price) before Christmas as well as alert the inhabitant to potential transformations of his habitat.

The second direction presupposes the architect to be the prime interpreter between physical form and human needs. The machine's role in this case is to exhibit alternatives, discern incompatibilities, make suggestions, and oversee the urban rights of individuals. In the nature of a public service the architect-machine partnership would perform, to the utmost of each actor's respective design intelligence, the perpetual iteration between form and criteria. The two directions are not exclusive; their joint enterprise is actually one.

What needs to be articulated, regardless of the format of the man-machine relationship, is the goal of humanism through machines. The question is not one of rationalism versus vitality (Juenger, 1949), nor the degree of rationalism (Ellen Berkeley, 1968), nor the castration of spirit by technique (Mumford, 1967). The concern is to avoid dehumanizing a process whose aim is definitely humanization. It is simply untrue that "unpleasant as it may be to contemplate, what probably will come to be valued is that which the computer can cope with—that is, only certain kinds of solutions to social problems" (Michael, 1963). We will attempt to disprove the pessimism of such comments. To do this, we will ask machines not only to problem-solve but also to problem-worry (S. Anderson, 1966).

In this book, there is no distinction between hardware and software, between special-purpose computers and general-purpose computers. The lines between what has been done, what can be done, and what might be done are all fuzzy. Our interest is simply to preface and to encourage a machine intelligence that stimulates a design for the good life and will allow for a full set of self-improving methods. We are talking about a symbiosis that is a cohabitation of two intelligent species.

001

Architect-Machine Symbiosis

Prelude to an Architect-Machine Dialogue

Something essential to man's creativity, even in science, may disappear when the defiantly metaphoric language of poetry gives way completely to the denatured language of the computer.
Lewis Mumford, *The Myth of the Machine*

You are in a foreign country, do not know the language, and are in desperate need of help. At first your hand movements and facial expressions carry most of your meaning to the silent observer. Your behavior uses a language of gestures and strange utterances to communicate your purpose. The puzzled listener searches for bits of content he can understand and link to his own language. You react to his reactions, and a language of pantomime begins to unfold. This new language has evolved from the mutual effort to communicate. Returning to the same person a second time, let us say with a new need, the roots of a dialogue already exist. This second conversation might be gibberish to a third party brought into the exchange at this time.

A designer-to-machine introduction should have a similar linguistic evolution. Each should track the other's design maneuvers, evoking a rhetoric that cannot be anticipated. "What was mere noise and disorder or distraction before, becomes pattern and sense; information has been metabolized out of noise" (Brodey and Lindgren, 1967). The event is circular inasmuch as the designer-machine unity provokes a dialogue and the dialogue promotes a stronger designer-machine unity. This progressively intimate association of the two dissimilar species is the symbiosis. It evolves through mutual training, in this case, through the dialogue.

Such man-machine dialogue has no historical precedent. The present antagonistic mismatch between man and machine, however, has generated a great deal of preoccupation for it. In less than a decade the term "man-machine communication" has passed from concept to cliché to platitude. Nevertheless, the theory is important and straightforward: in order to have a cooperative interaction between a designer of a certain expertise and a machine of some scholarship, the two must be congenial and must share the labor of establishing a common language. A designer, when addressing a machine, must not be forced to resort to machine-oriented codes. And in spite of computational efficiency, a paradigm for fruitful conversations must be machines that can speak and respond to a natural language.

With direct, fluid, and natural man-machine discourse, two former barriers between architects and computing machines would be removed. First, the designers, using computer-aided design hardware, would not have to be specialists. With natural communication, the "this is what I want to do" and "can you do it" gap could be bridged. The design task would no longer be described to a "knobs and dials" person to be executed in his secret vernacular. Instead, with simple negotiations, the job would be formulated and executed in the designer's own idiom. As a result, a vibrant stream of ideas could be directly channeled from the designer to the machine and back.

The second obstruction overcome by such close

This photograph first appeared in Edward Steichen's *The Family of Man*. (Photograph courtesy of Peter Moeschlin)

communion is the potential for reevaluating the procedures themselves. In a direct dialogue the designer can exercise his proverbial capriciousness. At first a designer may have only a meager understanding of his specific problem and thus require machine tolerance and compatibility in his search for the consistency among criteria and form and method, between intent and purpose. The progression from visceral to intellectual can be articulated in subsequent provisional statements of detail and moment-to-moment reevaluations of the methods themselves.

But, the tête-à-tête must be even more direct and fluid; it is gestures, smiles, and frowns that turn a conversation into a dialogue. "Most Americans are only dimly aware of this silent language even though they use it everyday. They are not conscious of the elaborate patterning of behavior which prescribes our handling of time, our spatial relationships, our attitudes towards work, play, and learning" (Hall, 1959). In an intimate human-to-human dialogue, hand-waving often carries as much meaning as text. *Manner* carries cultural information: the Arabs use their noses, the Japanese nod their heads. Customarily, in man-machine communication studies, such silent languages are ignored and frequently are referred to as "noise." But such silent languages are not noise; a dialogue is composed of "whole body involvement—with hands, eyes, mouth, facial expressions—using many channels simultaneously, but rhythmized into a harmoniously simple exchange" (Brodey and Lindgren, 1968).

Imagine a machine that can follow your design methodology and at the same time discern and

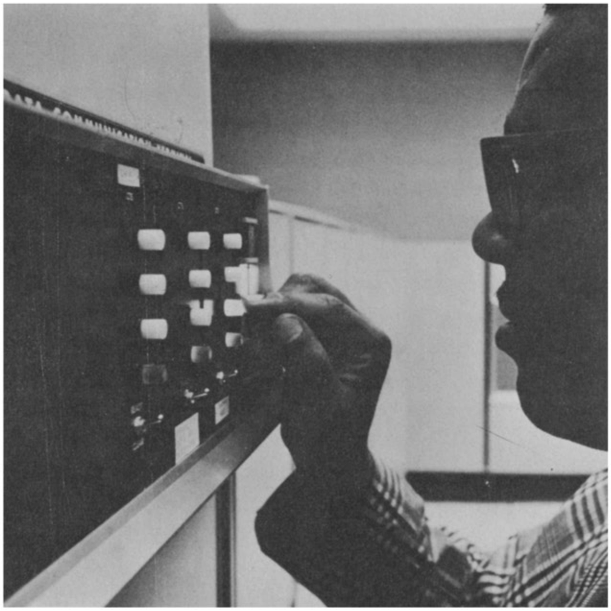
The sequence of photographs is taken from the 16mm film, *Three Experiments in Architecture Machines,* first shown at the Environmental Design Research Association Conference, Chapel Hill, North Carolina, June 1969. The prints are cropped from every fourth frame of a four-second scene. In these few seconds the user of this terminal has said more to the machine in hand-movement language than in any string of text, but it is all unheard. This particular person has never used a machine before; he does not know what a language is without gestures.

assimilate your conversational idiosyncrasies. This same machine, after observing your behavior, could build a predictive model of your conversational performance. Such a machine could then reinforce the dialogue by using the predictive model to respond to you in a manner that is in rhythm with your personal behavior and conversational idiosyncrasies.

What this means is that the dialogue we are proposing would be so personal that you would not be able to use someone else's machine, and he would not understand yours. In fact, neither machine would be able to talk directly to the other. The dialogue would be so intimate—even exclusive—that only mutual persuasion and compromise would bring about ideas, ideas unrealizable by either conversant alone. No doubt, in such a symbiosis it would not be solely the human designer who would decide when the machine is relevant.

The overlaying of a specific design character upon a generalized machine is not fanciful; subsequent chapters will illustrate some primitive attempts. An anonymous machine, after identifying a speaker, can transform itself into an exclusive apparatus that indeed would reflect previous encounters with that speaker. The extent of the metamorphosis depends on the degree of acquaintance. At the onset of the partnership, the machine gathers gross features; later it avails itself of subtleties. The design dialogue is one of mutual development.

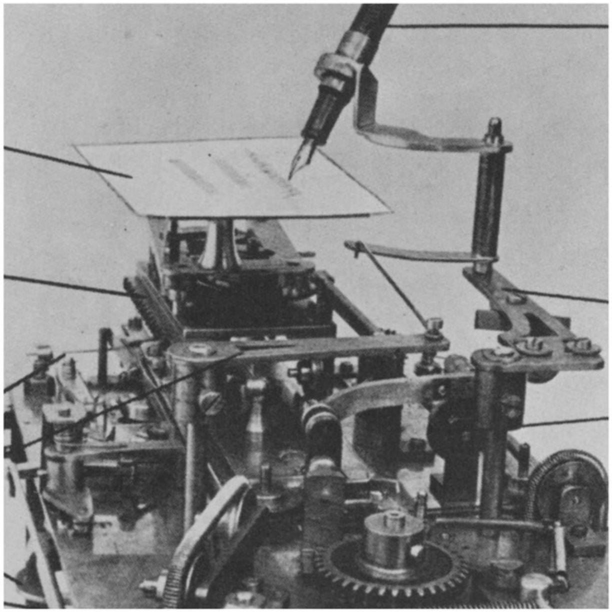
One might argue that we are proposing the creation of a design machine that is an extension of, and in the image of, a designer who, as he stands, has already enough error and fault.



The three photographs were taken by George DeVincent and first appeared in the *Institute of Electrical and Electronics Engineers Spectrum,* September 1967. The illustrations describe a succession of interfaces from hard to soft, from inanimate to animate. In most machines the interface where the flesh hits the steel (as Warren Brodey would say) is no more subtle and no more advanced than that of an old typewriter. (Photographs courtesy of the Environmental Ecology Laboratory, Boston, Massachusetts)

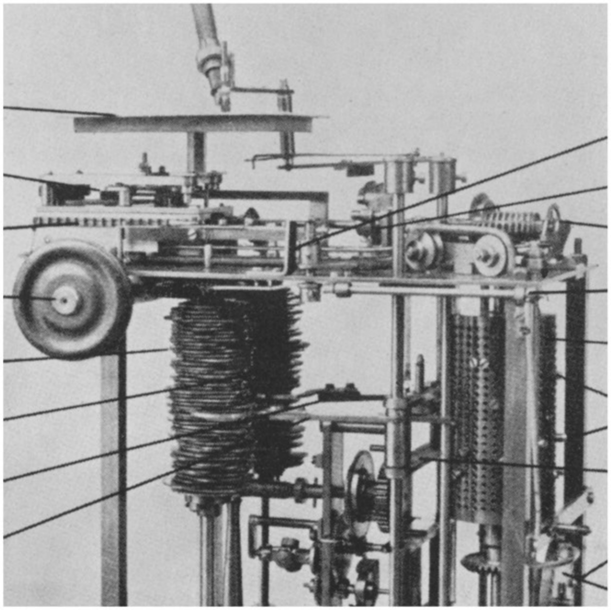
However, we have indicated that the maturation would be a reciprocal ripening of ideas and ways. At first, jobs where the man is particularly inept would stimulate a nontrivial need for cooperation. Subsequently each interlocutor would avoid situations notably clumsy for his constitution, while prying into issues that were originally outside the scope of concern (or the concern of his profession). Eventually, a separation of the parts could not happen: "The entire 'symbiotic' system is an artificial intelligence that cannot be partitioned" (Pask, 1964).

In the prelude to an architect-machine dialogue the solidarity of the alliance will rely on the ease of communication, the ability to ventilate one's concerns in a natural vernacular, and the presence of modes of communication responsive to the discipline at hand. A wine taster would expect his partner to have taste buds and an understanding of vintages. An architect would expect his associate to have at least a graphic ability capable of manipulating and displaying a host of environmental data and, in particular, physical form.



Writing machine made by M. F. Weisendanger. This device was actually built in 1946. When the mechanism worked, the amateur mechanician added, "People would be astonished to see a man of our time sacrifice so much leisure and so many hours to such a useless piece of work."

The device was built after studying the complete papers describing the Jaquet-Droz Writer, built in 1774. (Photograph courtesy of Editions du Griffon, Neuchâtel, Switzerland)



Natural and Not-So-Natural Computer Graphics

Man's prolific need for graphic expression can be seen in telephone booths, subway stations, and public men's rooms. More constructively, graphic media have been indigenous to architects. Traditional applications range from the thumbnail sketch to the rendering to the working drawing. In general, the conveniences of two-dimensional graphic representation have warranted overcoming the technical difficulties of describing three-dimensional events; consequently, mechanical drawing has become the "Latin" of all architecture students.

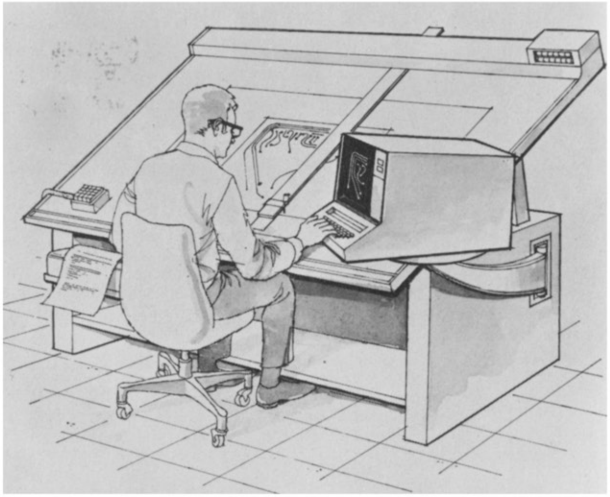
Now machines can do mechanical drawing too. So-called computer graphics has popularized the architect-machine dialogue by affording a natural language—the picture—where the designer can talk to the machine graphically and the machine can graphically respond in turn. This congenial technique is surely a natural way for architects to express their thoughts and is certainly in vogue. In the past few years, however, it has so dramatically overstated itself that the "message" has indeed become dominated by the "medium."

Computer graphics is not a synonym for computer-aided design. The significance of graphic interaction can be no greater than the meaningfulness of the content in the transaction. No matter how fancy and sophisticated the computer graphics system, it is only a glorified blackboard or piece of paper (even though possibly three dimensional), that is, until it overtly "talks back" and actually participates in the dialogue. Nonetheless, let us isolate computer graphics for a moment and look at it as a medium of communication.

There exist two families of graphic mechanisms: those devices used to "input" information to the machine and those for the machine to "output" information to the designer. One particular output mechanism of prime importance is the cathode-ray tube, a televisionlike display device. An electron beam, positioned by the computer, sweeps across the face of the scope (in an "on" or "off" state) to draw a picture by exciting tiny phosphors that glow for about a twentieth of a second. Once traced, the image is regenerated and continually redrawn on the face of the screen until a change in content imposes a recalculation of the beam's path. This regeneration is costly because, in order to deliver the illusion of a still image, it must occur between twenty and forty times per second, depending on the complexity of the picture.

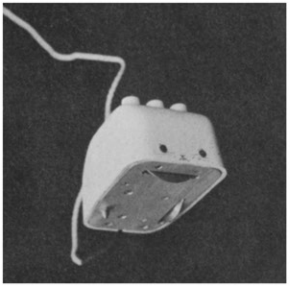
The cathode-ray tube's most common input device is the light pen. Rather than squirt out light, this stylus is a sensing device that can discern the light of the electron beam. With this instrument the designer can either detect lines, points, or characters, or he can drag about a spot of light, a tracking cross, to draw lines. At present it is not much like a pencil; it is a blunt pointer and to write with it is like applying a crayon to a postcard. The picture is small, the lines are thick, and the complexity of the displayed image is limited. Nonetheless, at present it is one of the more acceptable vehicles for research and does allow the necessary, real-time graphic intercourse.

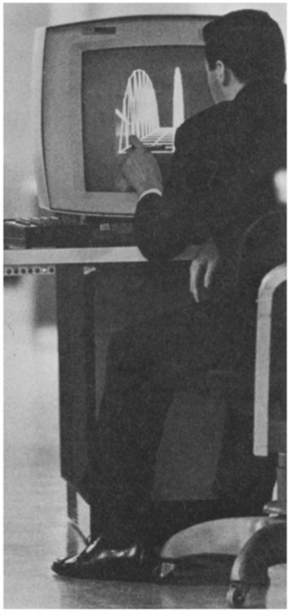
The awkwardness of display devices such as the cathode-ray tube goes beyond clumsiness. For example, one original acclaim in computer graphics was that "crooked lines are automat-


1


2


3


4


5


6

1 Computervision's INTERACT - GRAPHIC. Presently under development, this terminal combines several low-cost facilities into one configuration that will allow a high level of interaction. The unit is designed as a transition between present methods and future computer graphics. With this device the operator can even use his own pencil.

2 Computer Displays' Advanced Remote Display Station (ARDS). This three-faced configuration was designed for the Department of Architecture at M.I.T. Each screen is a storage tube, a device that will retain an image on the face of the scope without retracing with the electron beam. The scope does not allow dynamic displays (rotation, translation, etc.) and does not allow erasing parts of a picture without recreating the whole image. However, the unit requires very little computing in communication and costs less than 10 percent of an IBM 2250.

3 The Stanford Research Institute terminal used in the Augmented Human Intellect Research Center. The scope is a commercial (875 line) television monitor.

4 A mouse, used on both the Stanford Research Institute terminal and the ARDS. This mechanism is an input device, a cheap device ($400), and a clumsy device.

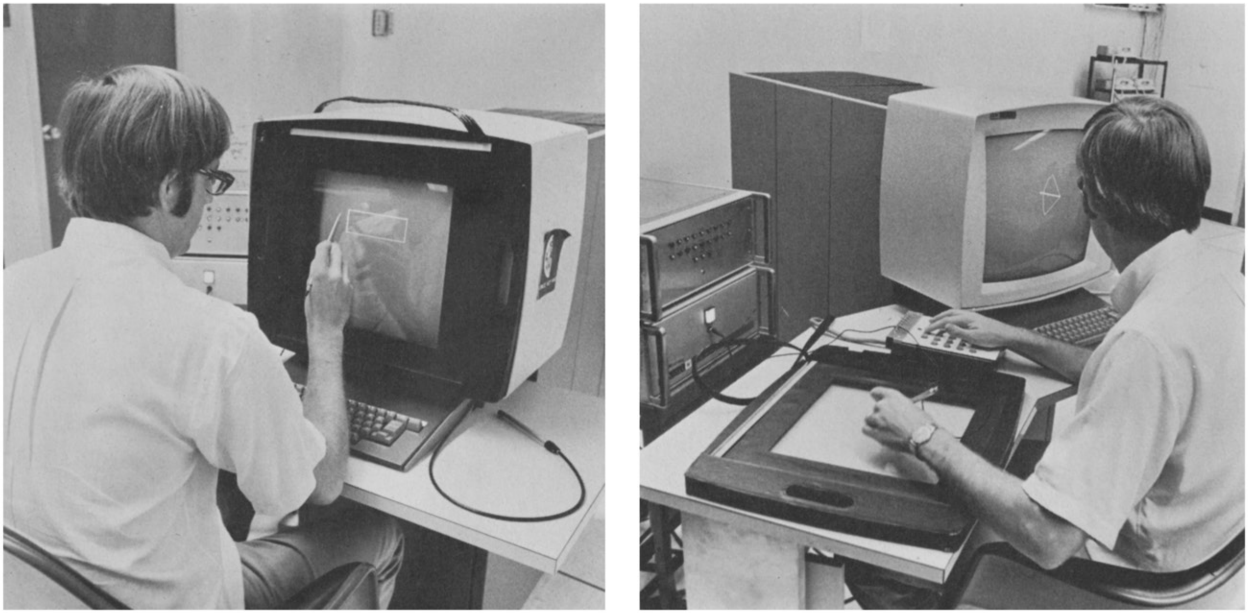
5 The IBM 2250. (Photograph courtesy of the IBM Corporation)

6 The Adage display unit.

ically turned into straight ones" (and if properly programmed, can even make them perfectly horizontal or vertical to the nearest millionth of an inch). Unfortunately, "instant accuracy" is not always desirable. In a design dialogue the wobbliness of lines often expresses the degree of clarity of architectural thought. The embodiment of an idea should reveal and be congruous with the stage of the design. One does not sketch with a 6H pencil and a straightedge or make working drawings freehand with a felt pen. The refinement of a project is a step-by-step process of sharpening both the comprehension and representation of one's image of the problem. A straight-line "sketch" on a cathode-ray tube could trigger an aura of completeness injurious to the dialogue as well as antagonistic to the design.

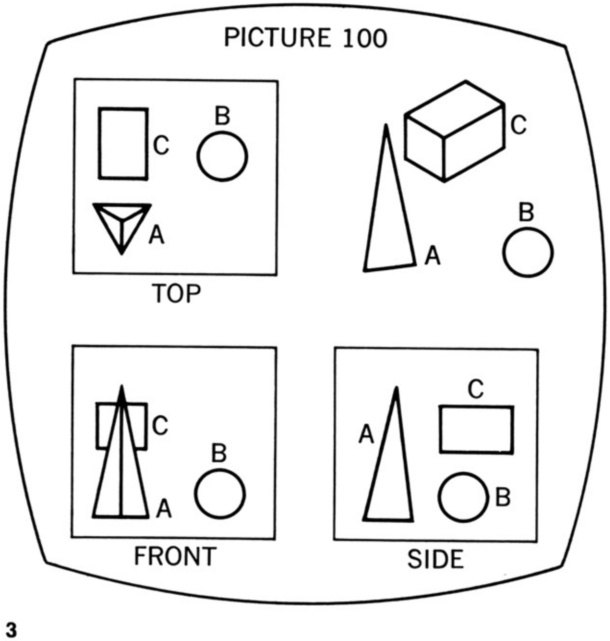
The clumsiness of computer graphics hardware is surrounded with technical difficulties, and, even when tackled, its resolution will not yield the same textural feeling as graphite on paper. Computer displays will force a new doodle vernacular if they are to capture those original ideas that usually reside on the backs of envelopes. Displays will have to allow for hazy negotiations to be sloppily expressed. In the meantime the important work of Timothy Johnson (1963) satifies the research need for a "sketchpad."

Beyond the antisketch nature of our present computer sketch pads, there is a second awkwardness. Traditionally, the architect has drawn plans, sections, elevations—two-dimensional representations—to describe graphically to himself and others his three-dimensional vision of an architectural solution. From the two-



1

2



3

dimensional documents, a three-dimensional representation, a physical model or perspective drawing, can be extrapolated. More recently the design process has been inverted in that we sketch with study models of clay, cardboard, styrofoam, or little wooden blocks. (Unfortunately, the gestalt of the forms generated by these three-dimensional study models unconsciously implies the form of the final solution.) In the later stages of design, sections are derived from the model in order to study or represent aspects concealed by, or unrepresentable in, the physical model.

In computer graphics, unlike the traditional trends and more like contemporary methods, a model always exists. Regardless of how it is stored within the machine, a description of the physical form must reside in the memory. From this internal description the machine can produce a section at any point, innumerable plans, and unlimited perspectives. Though it affords prolific two-dimensional output, this internal model becomes an imposition on the dialogue. For example, when drawing a section every point must have a clearly identified depth, or else the designer must draw in several orthogonal views simultaneously. Furthermore, the designer must explicitly tag surfaces and volumes. At their present stage of development computer graphics systems demand an a priori knowledge of whether the designer is working with lines, planes, or volumes, because each requires a different reception.

In computer graphics systems the architect is obliged to work in a predetermined mode (usually volumetric) which employs predefined elements whose proportions and scale may be

1 The Rolls Royce of displays, the IBM Cambridge Scientific Center's 2250, model 4, with Sylvania tablet. This configuration has a small computer (an IBM 1130) devoted to maintaining the graphics. The Sylvania tablet has been added to give both a smoother and a more simple way of drawing "into" the computer. The tablet is transparent as well as sensitive to the third dimension, in that it can recognize three discrete pen distances away from its surface (up to about one inch). The tablet can be used on the face of the screen (thus coincident with the displayed lines) as well as horizontally, off to the side.

2 Drawing by Morse Payne of The Architects Collaborative made on the IBM Cambridge Scientific Center's 2250 and subsequently plotted on a Calcomp plotter. This drawing displays a sketchiness that is most often absent in computer displays. It is composed of tiny lines whose end points are stored in the 1130's memory. Note that, at about the shoulder and foot, the 1130 ran out of memory locations and was unable to display the complete drawing.

3 The typical mechanical engineering format of top, front, and side view used in Timothy Johnson's SKETCHPAD III. By drawing in several views the machine is never confused as to where the lines belong, but the operator is. (Drawing courtesy of *IBM Systems Journal)* manipulated. Such a system was developed by Lavette Teague (1968) when at M.I.T. Teague's system—BUILD—allows the multiple juxtaposition of parallelepipeds. Spaces are described by volumes and are attached to each other by complete or partial surface-to-surface connections. In this case the topology of the shapes is kept constant, and the proportions are manipulated. The systems try to offer comprehensive, architectural computer graphics. It does not provide for a dialogue. It is computerized.

Computer-Aided versus Computerized
"Computerized" operations are too often misnamed "computer-aided." The computerized/computer-aided distinction is too often confused with, or solely embodied in, the mode of machine usage.

The traditional (for the past 20 years) mode of computer usage, "batch processing," entails a computation center to which a user delivers a "program" (a deck of cards, magnetic tape, paper tape) to be "run." Then several hours or days later the user returns to receive his "output." More recently, a new mode, "time-sharing," allows terminals (usually teletypes) in the office or at home. The terminals are connected to a large central machine (and thus interconnected with each other) by standard telephone lines. This system of remote and multiple machine access permits many physically separated users to share one large machine at the same time. The rapid swapping of users' programs in and out of the central machine provides each user with the illusion of a dedicated machine and permits him continual use of his terminal. This mode of operation is a form of "on-line" usage.

It is commonly suggested that by furnishing a time-sharing system the on-line nature of the interaction in itself is a dialogue and transforms computerized procedures into computer-aided ones. This is simply not true. For example, let us suppose you desire the average apartment-to-parking-space distance for some design project. In a batch-processing mode (assuming the program exists) you supply as data the description of your design, and the average distance returns hours later, indeed a computerized procedure. On the other hand, in a real-time environment you have a teletype terminal, the project description resides in the machine, and you simply type in the apartment-to-parking-distance command. But just because the answer comes back in three seconds rather than three days, computerized does not become computer-aided. It simply becomes more convenient "computerizedness." Computer-aidedness demands a dialogue; events cannot be merely a fast-time manifestation of causes and effects.

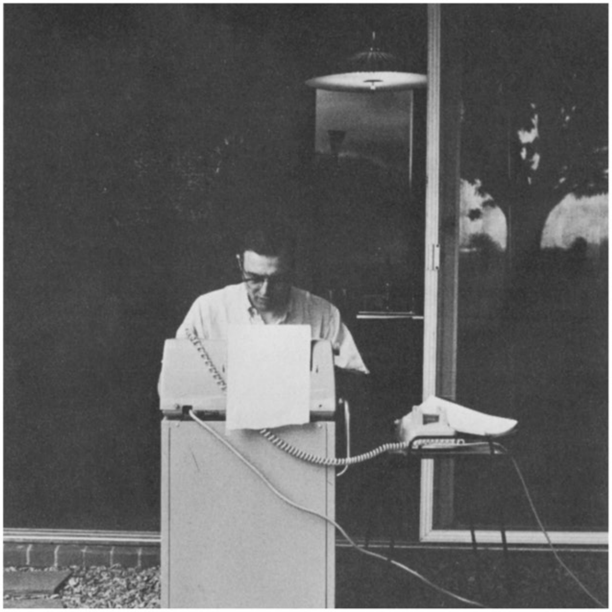
On-line communication therefore is not a sufficient (though necessary) condition for a computer-aided environment. Computer-aided design requires at least three additional features: (1) mutual interruptability for man and for machine, (2) local and dedicated computing power within the terminal, and (3) a machine intelligence.

Interruptability gives a dimension of interaction that allows the process, as well as the product, to be manipulated. In a computer-aided system, the machine may interrupt the user and present the unsolicited information, for example, that the cost of his low-income housing project is fifty-eight dollars per square foot. The architect might welcome the remark, ignore it, or take offense and request that such interludes of finance be restricted. However, regardless of the designer's response, the apparent high cost might have overlooked substantial indirect savings not accounted for in the original estimating routine. In this case the designer could tamper with the estimating procedure and incorporate hitherto neglected parameters.

Unfortunately, the present time-sharing philosophy fosters a cause-and-effect conversation. Time-sharing assumes that a designer's explicit manipulations will occupy between one and ten percent of any sitting; the remaining time represents his deliberations and distractions. Each user's moments of contemplation are in effect another user's instants of computation. A designer can interrupt his own program, but a routine cannot easily interrupt its partner in thought. In order to leave the computational utility available for other users, each routine resides in the machine only when explicitly called into service by its particular user. In other words, the routine (the user's machine) can listen but cannot interrupt.

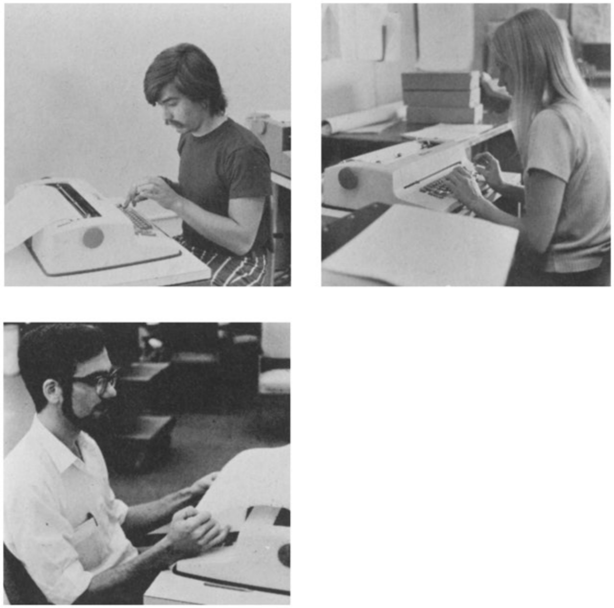
To retain the assets of time-sharing, avoid the anathema of batch-processing, and acquire mutual interruptability, we adjust the allocation of computing power. We transfer some of the information-processing power and transfer a certain manipulative and storage capacity to the terminal that was originally a teletype transmission and reception device. This semiautonomous terminal (possibly portable) is a small computer that would be a "machine in residence." An architecture machine would be such a machine. The designer would speak directly to this satellite machine. In turn, this small, remote computer would interactively converse with larger parent machines. (Sending work out to a central mechanism would be automatic and exclusive of the designer; the recourse would be for reasons of speed or memory or information or all three.)

The machine at the location of the designer would undergo the personalization. It would be


1


2


3


4

1 Leon Groisser at home in his garden.

2 The author at home.

3 Computers at home are already being used in an informal manner.

4 Architecture students using the time-sharing system CP/CMS. Since 1965, all M.I.T. architecture students have been required to take at least one semester of computer programming as a prerequisite to the Bachelor of Architecture degree. Most of them have had the good fortune to learn on a time-sharing system. The advantage is obvious: on a console, a student can take high risks and can play. This is what learning is all about.

composed of additive and subtractive pieces of hardware as determined by the discipline of its partner. This local aggregation of parts would perform the dialoguing, the evolving, and the interrupting. Observe that the interrupting and the reinterrupting would depend on the nature of the designer's activities, on the context of his efforts. Through familiarity with a specific designer's idiosyncrasies, the appropriateness of the machine's interruptions would be suitably reinforced by context—the inception of an intelligent act.

A mechanical partner, as we have suggested, must have intelligence. Customarily, computer-aided design studies and intelligent automata studies have been antipodal efforts "between mechanically extended man and artificial intelligence" (Licklider, 1960). On the one hand, in the context of computer-aided design we are told to render unto each their respective design functions and talents: man thinks and the machine calculates. On the other hand, in the context of automata studies we are told that "Anything you can do, a machine can do better."

The two outlooks are not necessarily contradictory. For the present discussion there is a real issue whether machine intelligence can be independent of human intelligence. In computer-aided design only the combination of mechanical amplification and mechanical imitation will validate the dialogue. The dialogue will evolve an intelligence, this intelligence will stimulate a more profound dialogue, which in turn will promote further intelligence, and so on. Furthermore, the concurrence of "extended designer" and "artificial designer" will force a design redundancy and an overlapping of tasks that are necessary for the understanding of intricate design couplings. Perpetual cross-examination of ideas by both the man and the machine will encourage creative thought that would otherwise be extinguished by the lack of an antagonistic (and thus challenging) environment. Computer-aided design concerns an ecology of mutual design complementation, augmentation, and substitution.

Adaptable Machines, Sensory Machines, and Parent Machines

A computer-aided design system is too often characterized or glorified by its size and its repertoire of operations. A zoo of design services frequently provides the designer with the illusion of generality through sheer quantity of specific routines. In Steven Coons's original *Outline of the Requirements for a Computer-Aided Design System* (1963), the danger of exhibiting a false generality has been well marked. As long as the designer never calls for a capacity that is not rigidly embedded in the machine, the system will be successful. However, since it is not feasible to predefine and to pinpoint all plausible operations and design activities, it follows that a successful design partner might be composed of one intelligent and adaptable service rather than a group of special-purpose services.

The principle is simple and, in computer-aided design history, old. A well-nourished platoon of specific design operations expects a status quo and excludes a methodological evolution and self-improvement. As a consequence, so-called problem-oriented languages have been developed in an attempt to avoid this stagnation by providing each user, after a brief learning period, with the potential of creating his own tailor-made utilities.

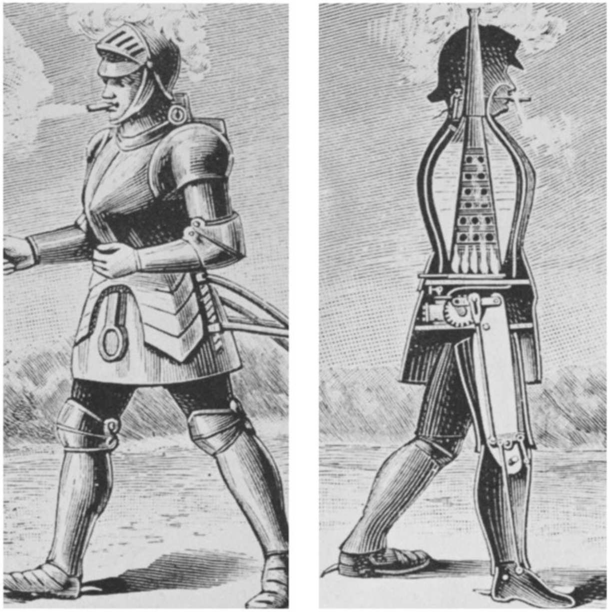
A problem-oriented language is a high-level computer language whose formulation and implementation assume a specific discipline or set of disciplines. Such a language provides the equivalent of a set of nouns, verbs, and phrases. A user can easily learn them because of their simplicity and relevance to his profession. For a civil-engineer user, basic operations, like calculating bending moments or shear forces, might appear as verbs and be combined with declarations; for example, TYPE PLANE TRUSS YZ/LOADING LIST 'TRUS-UNI'/DETERMINATE ANALYSIS (Logcher, 1967). With such commands, the user can implement his own algorithm for determining the behavior of a structure.

But two things are wrong. First, we have a condition where each designer is creating his own library of services out of the problem-oriented language. Once created, note that these operations are no less rigid and specific than the predefined package of design commodities. Even though the routines are user chosen and user made, they might be less effective than if created by someone (or a machine) versed in the computer sciences, with the full potential of lower-level languages available to him. Second, when using a problem-oriented language, the user-made repertoire of operations is largely determined by the language itself and the user's understanding of it rather than by the nature of the design problem itself. The appearance of particular commands in the language and the absence of others completely prejudices both the choice of problem and the method of implementation. In other words, a problem-oriented language gives the same illusion of generality as the rigid regiment of services. The common failure is a misunderstanding of the difference (which is not a semantic difference) between flexibility and adaptability.

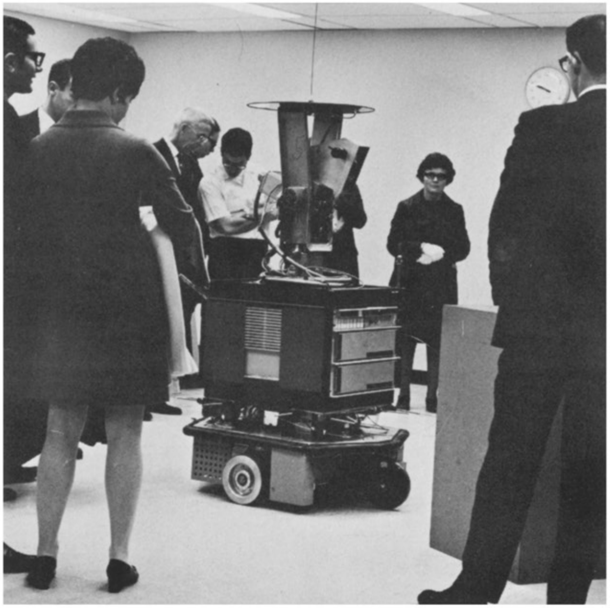
The omission is evolution. A dialogue must be evolutionary; a mechanical partner must be evolutionary and hence adaptable. An adaptable machine is a generalized mechanism that at any instant can transform itself (in response to a change in context) to appear as a special-purpose machine. By sampling its environment, an adaptable machine could freely move from a state of universality to a state of single-mindedness.

No adaptable machine exists today. However, we can (and should) discuss the environment that such machines might sample in order to transform themselves. So far we have presented a duet—designer and machine—in which the machine's "image" of the real world is solely through one human partner. The designer's personal prejudices and distortions of the real world would be planted, consciously or subconsciously, in the machine. In such a closed system the machine could easily develop into a "design patsy" or "yes-man." The machine would not challenge goals; it would only be prepared to mimic the communicative manner and methodology of its one user. In this situation the designer could embed his preconceived answers, and, accordingly, a noncreative, complacent partnership would be formed through the lack of a challenging environment.

Beyond the one-architect-one-machine dialogue, the milieu of an adaptable machine must embody two further contacts with the real world. First, an adaptable machine (and thus an architecture machine) must receive direct sensory information from the real world. It must see, hear, and read, and it must take walks in the garden. Information should pass into the machine through observation channels that are direct rather than undergo the mutations of transfer from the real world to designer's sensors to designer's brain to designer's effectors



1



2

1 George Moore's steam man (circa 1893). A gas-fired boiler operated an engine of ½ horsepower. Exhaust was let out through the helmet, steam through the cigar. Its walking speed was between seven and nine miles per hour. (Photographs courtesy of Ronan Picture Library, Newmarket, England)

2 The Stanford Research Institute robot.

to machine's sensors. The designer does not completely control data that are collected in a manner that avoids the consequent losses of information at each transfer point. Such data bolster, once again, the machine's capacity to challenge.

A great deal of research is being conducted in the quest for mechanical sensors. Probably the first to be incorporated with an architecture machine will be a seeing machine; this will be briefly described in a subsequent chapter. At first, machines with eyes will observe simple physical models; eventually, they will observe real environments.

A second tie to the real world would be the capability of overviewing designers, their definitions, activities, and methods. Surveillance of other designers' procedures again nourishes the machine's ability to challenge. For example, the machine could alert its partner to the practices of other designers. (They provide greater direct outdoor access in high-density residential areas.) This response would allow the designer either to correct himself or to reinforce his own and his machine's convictions further. At the same time, each designer would be able to tune into controversy by dialing an anonymity or an opposing view that would discount his own whims and pet details. A designer would be able to subject himself and his machine to an objective scrutiny that would consist of (1) an evolutionary mapping of popular desires, (2) a statistical overlay of solution patterns, and (3) the images of architects he esteems.

This overviewing would be achieved through a parent machine, not by machines in residence. The same central machine that provides the big bursts of computational power would be host to all the local issues resulting from many separate architecture machines. The parent machine can be a referee, an information source, a communication medium, a historian, as well as simply a giant calculating mechanism. The parent machine would store all building codes, *Sweets Catalog, Graphics Standards,* and all the geographic and demographic data of the world. (The reader should note that the constitution and design of such a machine is not the prime concern of this book. Our main concern here is the machine in residence.)

An architecture machine that could observe existing environments in the real world and design behaviors from the parent would furnish the architect with both unsolicited knowledge and unsolicited problems. Someday machines will go to libraries to read and learn and laugh and will drive about cities to experience and to observe the world. Such mechanical partners must badger us to respond to relevant information, as defined by evolution and by context, that would otherwise be overlooked.

Machines that poll information from many designers and inhabitants, directly view the real world, and have a congenial dialogue with one specific designer are architecture machines. They hint at being intelligent machines.

010

Aspects of Design Procedures

From Perspectives to Holography

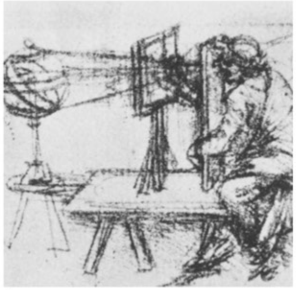
Drawing a perspective is a procedure for visualization, a procedure that has suffered from faddism in computer graphics, a procedure that can unfairly influence the process of creating physical form. Alberti's formalization of perspective in 1432 helped foster the Renaissance preoccupation for the observer and the viewing position (and possibly symmetry). Later, when man regularly built above six stories, the "bird's eye" and the "worm's eye" view made the stationary viewer even more manifest. Photography even further reinforced this syndrome.

Finally, the movie camera relieved the stationary-observer obsession by allowing the consideration of a path of movement and rotations of a field of vision. Cinematographic methods, however, were cumbersome, and the film processing time made movie-making a presentation procedure (off-line) rather than a study medium (on-line). Then came the instantaneous images of closed-circuit television. Coupled with a model scope or fiber-optics cord (optical devices for visually placing oneself within scale models), a designer could push his way through a model to simulate roughly the visual experience. Unhappily, television techniques are unwieldy.

The computer is a natural medium for the mass production of perspective images. At first, numerically controlled plotters were employed to draw perspectives at hundreds of small increments along a path. These drawings were then filmed with animation procedures to produce a cartoon of moving figures (Fetter, 1964), a general procedure more cumbersome than *any* previous method. Then the plotter was replaced with the cathode-ray tube in anticipation of creating perspective drawings (in their appropriate transformations) at a rate of sixteen to twenty frames per second, for providing the illusion of traveling through an environment at any speed and in a flicker-free manner (Negroponte, 1966). Assume that on the screen of a cathode-ray tube we have a perspective drawing, derived from the machine's model of some project. The rendered perspective is a crude jungle of lines describing a wire frame structure. Larry Roberts (1963) has taken out the hidden lines, David Evans et al. (1967) have put in halftones, and everybody is trying to perform the perspective transformations in real time.

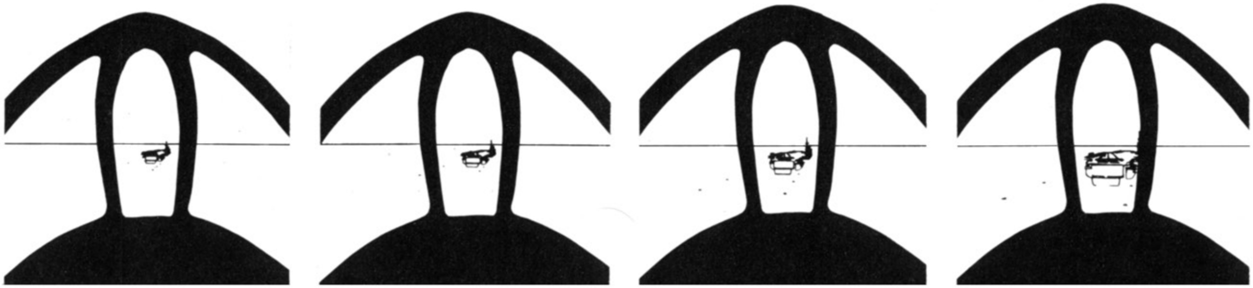
Meanwhile, General Electric's Electronics Laboratory, Syracuse, New York, under NASA contract, has developed a special-purpose computer that permits a viewer to voyage through an environment with hidden lines removed, with halftones, in real time, and in color. Furthermore, the user of this system commands the movement with an aircraft-type control stick that delivers him a motor involvement with the visual simulation. P. Kamnitzer of U.C.L.A. is presently applying the NASA-General Electric system to urban visual simulation problems (Kamnitzer, 1969).

The history is long; the list of participants is long (M. Milne, 1969). Why the great concern with perspectives? First, the problem is intrinsically natural for computer graphics studies, its formulation is technically difficult (thus stimulating), and it requires no examination of design philosophy.

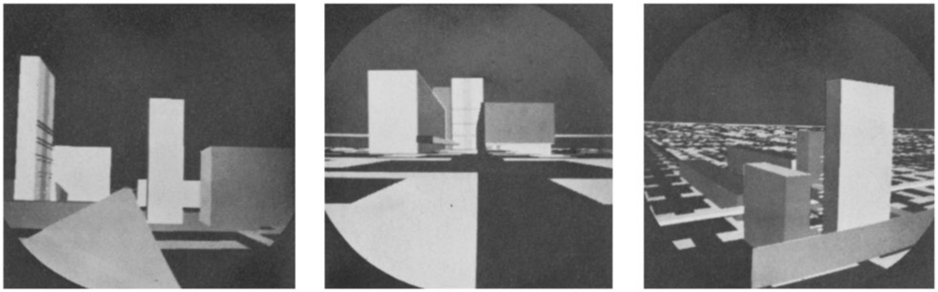


1

2

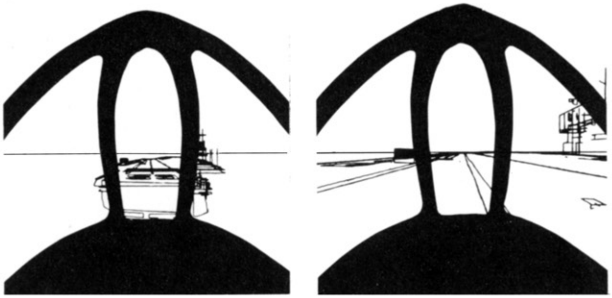


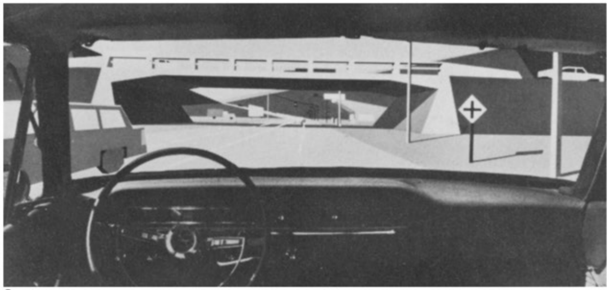
4



5

3





6

1 da Vinci: "Le Prospectographe," drawing circa 1488. (Courtesy of Biblioteca Ambrosiana, Milan)

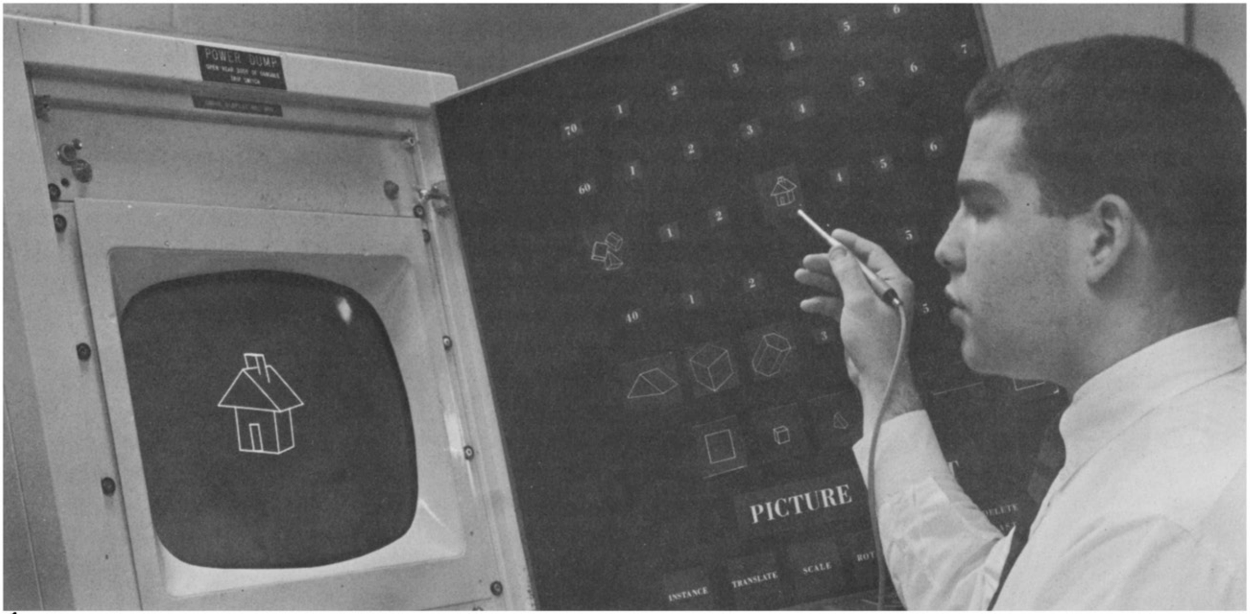
2 Durer: "Le Dessinateur de la femme couchée," engraving, 1525.

3 Durer: "Le Portraitiste," engraving circa 1525. (Courtesy of the Bibliothèque nationale, Paris)

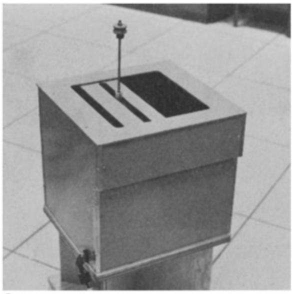
4 Six frames of a computer graphics film on visibility studies of an aircraft carrier landing. The film was made by William Fetter for the Boeing Company. The last ten seconds of the carrier landing required two hundred and forty computer-drawn perspectives to be plotted, touched up by an artist, and then filmed. (Illustrations courtesy of the Boeing Company)

5 Three black and white photographs of Peter Kamnitzer's color display at the Visual Simulations Laboratory of General Electric. The images are from the CITYSCAPE program, and they are presently restricted to 240 edges per frame. (Photographs courtesy of Peter Kamnitzer)

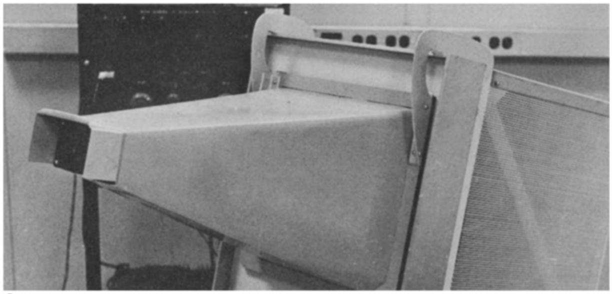
6 A rendering made to study the effects of increasing to 1,500 edges in the above system. (Courtesy of Peter Kamnitzer)



1



2



3

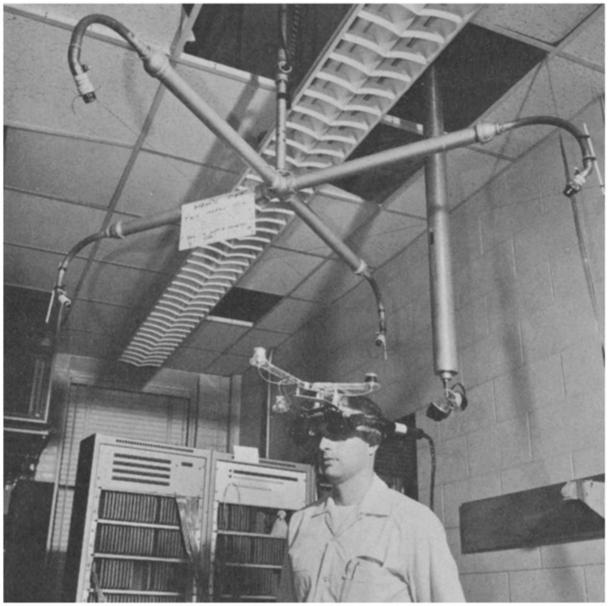
1 Larry Roberts' Wand. (Courtesy of Lincoln Laboratories)

2 An electromechanical device used for input of three-dimensional data. The device is much like an aircraft joy stick and is coupled with the adjacent stereoscope. (Photograph courtesy of Michael Noll)

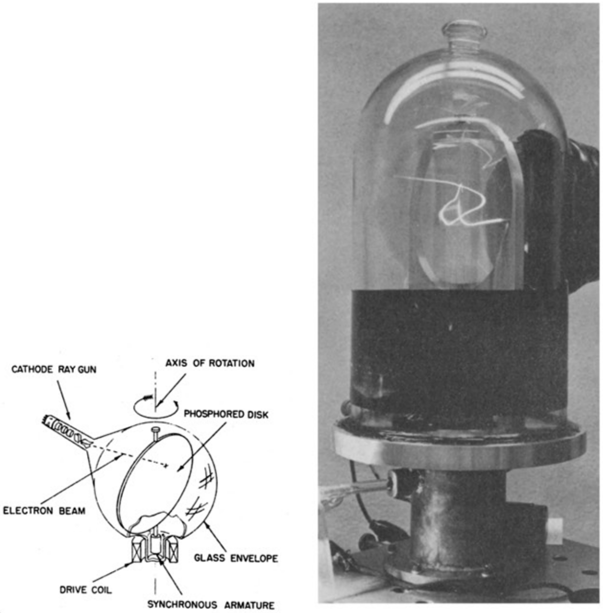
3 A stereoscopic viewing attachment on a large cathode-ray tube. This attachment was designed by C. F. Mattke for use by Michael Noll in his investigation of three-dimensional man-machine communication, performed at the Bell Telephone Laboratories. (Photograph courtesy of Michael Noll)

Perspective is a natural procedure for representing in two dimensions the illustration of a three-dimensional event. On a picture plane a trace of points defines the intersection of imaginary lines between a monocular observer and the real or unreal world. When the picture plane is removed from this world and viewed from the same vantage point, the image is an accurate representation with no distortion. The mode thus affords an appropriate visual representation of the visual aspects of an architectural real world. But, with future three-dimensional displays and input mechanisms, the virtuous role of the perspective drawing surely will be diluted. As Coons states, "In a few years from now (April 1968) you (a group of architects) will be able to walk into a room and move your hand and have a plane or surface appear before you in light. You will be able to build a building in light so that you can walk around it and change it" (Herzberg, 1968).

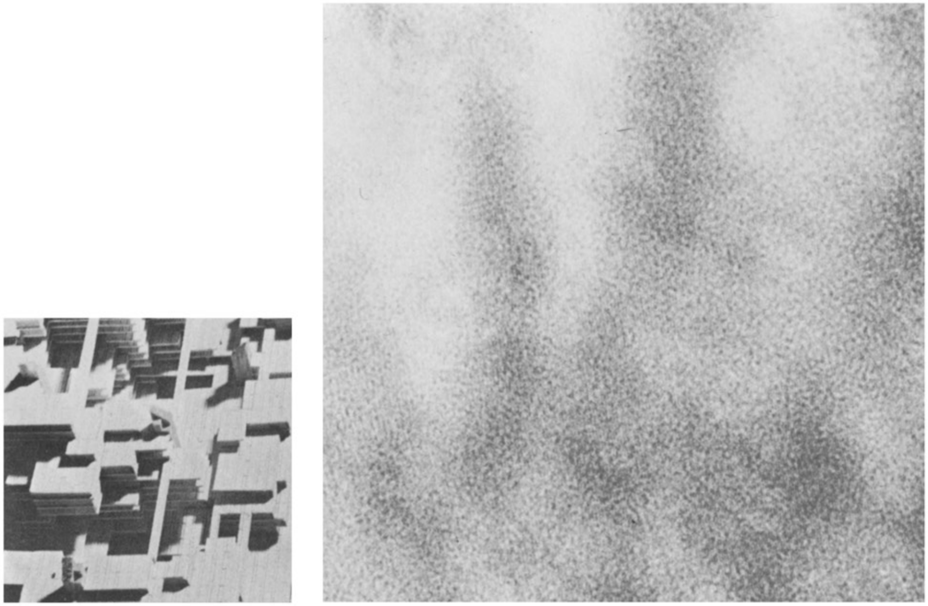
The dramatics of such dazzling statements stem from the age-old desire of the architect to be able to lift his pencil, gesticulate in mid-air, and have the stylus ooze out lines that float in space. Part of this desire has already been fulfilled by an ultrasonic position-sensing device. Larry Roberts' Lincoln Wand allows the computer (within a work space of ninety-six cubic feet) to track the *x-y-z* position (to the nearest one-fifth of an inch) of a hand-held, pen-size device (Roberts, 1966). Four ultrasonic transmitters recurrently pulse bursts of energy, and the Wand reports to the computer the time at which it heard each signal. The computer uses three time lengths to determine trigonometrically the pen's position; the fourth transmitter provides a geometric check on the



1


CATHODE RAY GUN
AXIS OF ROTATION
PHOSPHORED DISK
ELECTRON BEAM
GLASS ENVELOPE
DRIVE COIL
SYNCHRONOUS ARMATURE

2



3

1 Ivan Sutherland's helmet. (Courtesy of Ivan Sutherland)

2 The Direct-View Three-Dimensional Display Tube developed at Hughes Research Laboratories by R. D. Ketchpel (1963). In contrast to other presentations in which the third dimension is simulated stereoscopically, this device displays the information in actual space. The three-dimensional display tube utilizes a phosphor-coated disc spinning at 900 rpm. Upon excitation by a cathode-ray beam at selected times, any point in the volume "swept out" may be illuminated thirty times a second. (Photograph courtesy of R. D. Ketchpel)

3 This illustration is a direct positive print of a hologram of the model to the left. If viewed as a transparency with a coherent light source behind, this hologram would display the model in three dimensions.

measurements. Unfortunately, the Wand does not leave traces in midair upon which to build consecutive lines (which aggravates the problem of hand trembling). Though a three-dimensional model is being constructed within the machine, the output associated with the Wand is at present a perspective or axonometric display.

Further efforts will eventually allow three-dimensional displays to be joined with wandlike devices. Ivan Sutherland is creating a machine that gives the illusion of actually walking around and within visual models (I. Sutherland, 1968). The device is a helmet mounted with two eyeglass-size cathode-ray tubes (with prisms) that permit stereoscopic images to be transformed in accordance with the head position of the wearer. In this case three antennas report the user's position, but the movement could also be monitored with the user driving a simulated car and actually driving through a city that does not exist in the real world. With halftones, color, and real time, this technique would afford an excellent simulation of the visual world. Sutherland's device even allows for a split image, through the prisms, that permits the designer to view his project overlaid upon the visual real world.

Another three-dimensional display technique is holography (Gabor, 1948). *TV Guide* periodically tempts its readers with three-dimensional television: ballet dancers in your living room and the Tonight Show in your bed. In hologram television, "the pictures have a realism unattainable by any other means. The three-dimensional effect is obtained without the need for a stereo pair of pictures, and without the need for any devices such as Polaroid glasses. In addition, all the visual properties of the original scene, such as parallax between near and far objects in the scene, and a change in perspective as a function of the observer's viewing position, are present" (Leith and Upatnieks, 1965). This apparition is achieved by recording the interference patterns of two sources of coherent light (usually lasers), one reflected directly from the object and the other by a mirror.

At present, efforts are being conducted to construct through computation synthetic holograms for simple geometric configurations (Lesem et al., 1968). One method calculates the interference patterns and plots the result on a transparency. Another method positions a small mirror in three-dimensional space and traces the configuration in the presence of the necessary light sources, in effect, taking a time-lapse hologram (Stroke and Zech, 1966). So far, neither method is in real time.

When computers can simulate holograms in real time (using some flat-screen television technique), views of the machine's mathematical model could be selected in a general manner, and the designer's head movements could supply specific vantage points. Soon, on a display device, architects will have glimpses of physical environments that do not exist. These witnessings will be in full color, with halftones, and in three dimensions.

The reader must remember that these apparently ghostlike images are only visual simulations. Though the better ones will furnish a motor involvement with the designer, the devices that have been discussed do not delve into the crucial problem of machine response to nonvisual involvement with the environment: auditory simulation, tactile presence, feeling of a breeze in a lonely space.

Generation of Solutions

To some of the more hidebound architects the concept of a machine generating three-dimensional solutions is immoral, impossible, or endorses unemployment for threatened architects. The premise is that a human architect's experience gives him license to be the exclusive translator of human requirements into physical form.

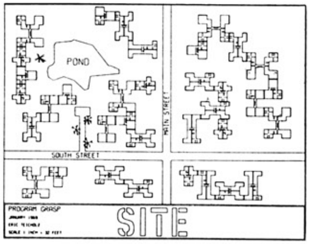
Physical form, according to D'Arcy Thompson (1917), is the resolution at one instant of time of many forces that are governed by rates of change. In the urban context the complexity of these forces often surpasses human comprehension. A machine, meanwhile, could procreate forms that respond to many hereto unmanageable dynamics. Such a colleague would not be an omen of professional retirement but rather a tickler of the architect's imagination, presenting alternatives of form possibly not visualized or not visualizable by the human designer.

An architect would not and should not confront a "criteria machine" to decrease visual privacy, increase public access, and watch contortions of form on a television screen. Instead, in the rhythm of the dialogue, a solution-generating capacity would be an evolutionary enterprise where the machine would act in "interrupt" or "reply" to its partner's activity. The architect might search for a configuration by observing the machine's attempts at satisfying a statement of the problem, or the machine might learn by observing the architect. In such a system both the architect and the machine could interrogate each other in order to locate those characteristics of the site and the criteria that imposed certain factors and courses of action on the generated solutions.

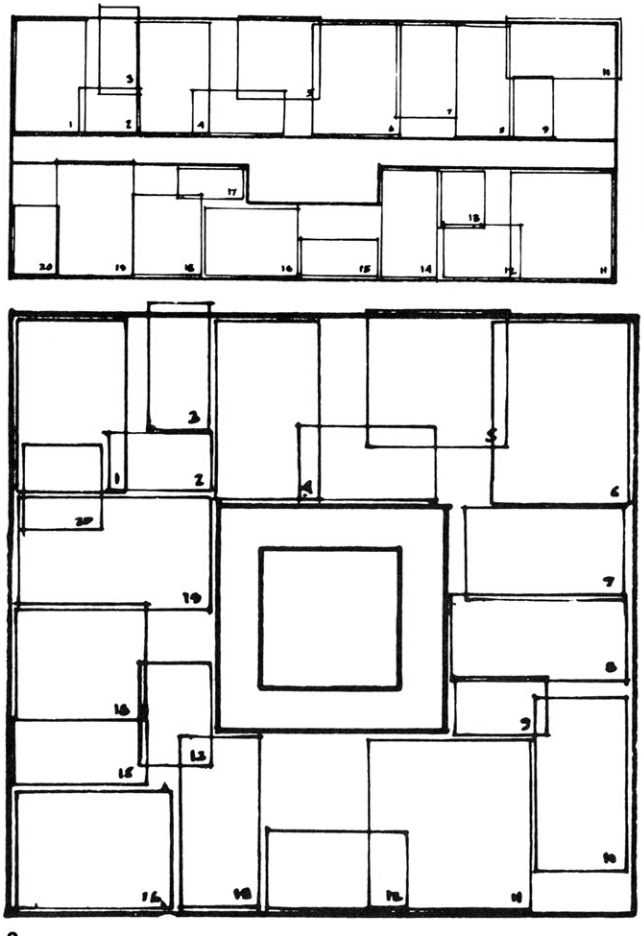
There are two distinct types of generated solution: one accommodates underconstrained problems; the other works within overconstrained situations. The underconstrained situation (rare in architecture) has a large set of possible solutions. The criteria are satisfied by many alternatives. These alternatives must then be evaluated by the architect using "intuitive" means, selection criteria he either does not understand or has never presented to the machine.

In the overconstrained problem, the generating mechanism is presented with great amounts of factional criteria that no form can completely satisfy. The generating mechanism searches for a solution that best relaxes the constraints, a point of greatest "happiness" and least "friction." The resulting form is a status of criteria compromise where the constraints least antagonize one another.

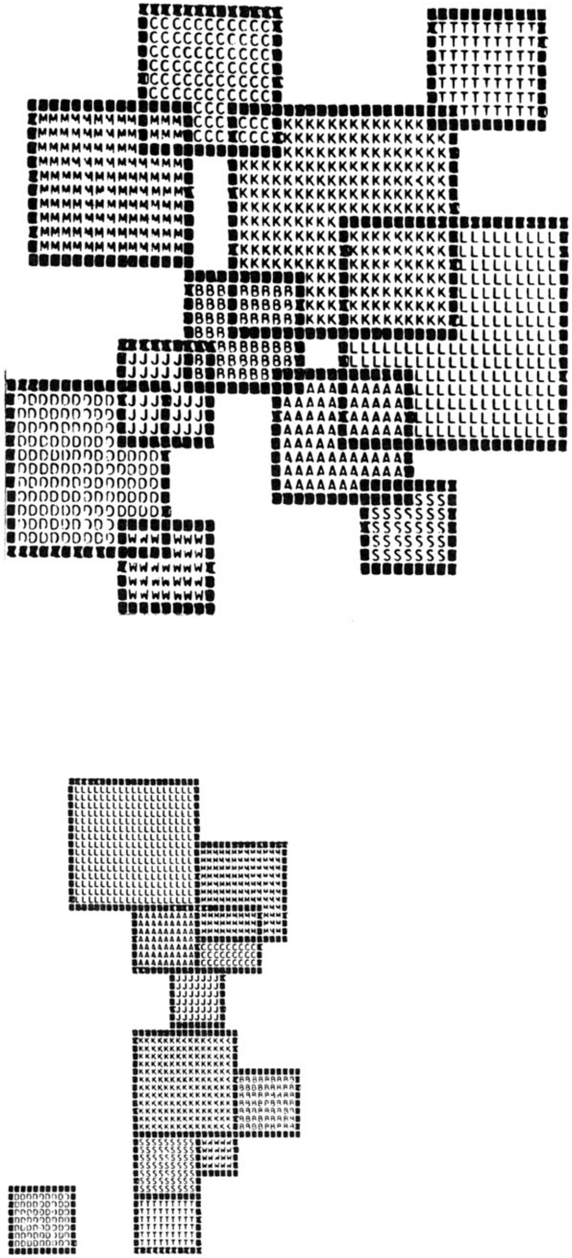
Both problem types involve trial-and-error procedures, tasks well suited for self-improving machines. In many cases random numbers are employed; they deserve mention, as their use is often misunderstood. In solution generation, a random number is a substitute for missing information or unpredictable information. Rather than just cast a Monte Carlo atmosphere of surprise, random numbers simulate nondeterministic events such as family displacements, employment changes, physical expansion. Usually, as a system grows, events become more and more deterministic, and the possible alternatives diminish. Generating pro-



1



2

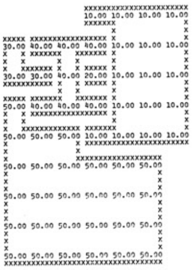


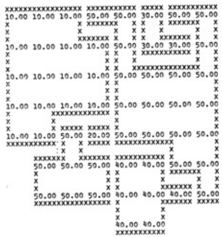
3

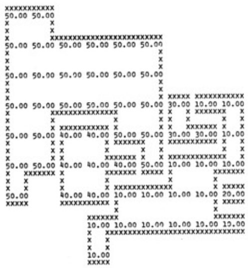
ALL

MEMBER NAMES	SQ FT	CST/FT	G/N	COST
PATIENT CARE SERVI	159500			$ 12948320
INPATIENT SERVICES	98500			$ 7538838
MEDICAL DEPARTMENT	0			$ 0
DERMATOLOGY	0	18	1.65	0
~~CARDIOLOGY~~	0	18	1.65	0
ALLERGY	0	18	1.65	0
~~RENAL DISEASES~~	0	18	1.65	0
HEMATOLOGY	0	18	1.65	0
~~ENDOCRINOLOGY~~	0	18	1.65	0
GENERAL MEDICINE	0	18	1.65	0
SURGICAL DEPARTMEN	0			$ 0
GENERAL SURGERY	0	38	1.65	0
~~ORTHOPEDIC SURGER~~	0	38	1.65	0
UROLOGY	0	38	1.65	0
~~THORACIC SURGERY~~	0	38	1.65	0
PROCTOLOGY	0	38	1.65	0
~~PLASTIC SURGERY~~	0	38	1.65	0
DENTISTRY	0	38	1.65	0
OTHER MEDICAL/SURG	0			$ 0
OPHTHALMOLOGY	0	38	1.65	0
~~OTOLARYNGOLOGY~~	0	38	1.65	0
PHYS MED & REHAB	0	38	1.65	0
~~PSYCHIATRY~~	0	38	1.65	0
NEUROLOGY/SURGERY	0	38	1.65	0
PEDIATRICS	20000			$ 2210998
NEWBORN NURSERIES	4000	67	1.65	442199
~~PREMATURE NURSERI~~	3500	67	1.65	386924
ISOLATION NURSERI	5000	67	1.65	552749
~~ADOLESCENT CARE~~	7500	67	1.65	829124
~~NURSING UNITS~~	65000			$ 3456743
~~GENERAL NURSING~~	37500	28	1.65	1732499
INTENSIVE CARE	5000	38	1.65	313499
~~CARDIAC CARE~~	1000	38	1.65	62699
PULMONARY CARE	0	38	1.65	0
~~PSYCHIATRIC CARE~~	5500	38	1.65	344849
METABOLIC DISEASE	0	38	1.65	0
~~SELF CARE~~	0	38	1.65	0
EXTENDED CARE	10000	38	1.65	626999
~~CLINICAL RESEARCH~~	4000	38	1.65	250799
BURN UNIT	2000	38	1.65	125399
SURGICAL SUITE	13500			$ 1871097
OPERATING ROOMS	11250	84	1.65	1559249
RECOVERY ROOMS	1500	84	1.65	207899
ANESTHESIOLOGY	750	84	1.65	103949
AMBULATORY CARE	28000			$ 1404972
EMERGENCY ROOMS	2500			$ 276374
EMERGENCY ROOM	2500	67	1.65	276374
OUTPATIENT CLINICS	21500			$ 996599
GENERAL MEDICAL	5000	20	1.65	164999
~~DIAGNOSTIC & CONS~~	2000	20	1.65	65999
SUGICAL CLINICS	6000	45	1.65	445499
~~SPECIALTY CLINICS~~	2000	25	1.65	82499
SPECIALTY PROGRAM	5500	20	1.65	181499
~~REHABILITATION &~~	1000	34	1.65	56099
~~MISC OUTPATIENT~~	400			$ 131999
~~HOME CARE~~	400	20	1.65	131999
COMMUNITY MENTAL	0	20	1.65	0
~~MULTIPHASIC SCREE~~	0	20	1.65	0
POISON CONTROL	0	20	1.65	0
AFFILIATED SERVICE	0			$ 0
HILLCREST	0	0	1.65	0
~~HEARING & SPEECH~~	0	0	1.65	0
CHILD HEALTH CENT	0	0	1.65	0
~~RETARDED CHILDREN~~	0	0	1.65	0
OTHER AFFILIATES	0	0	1.65	0
ADJUNCT DIAG/THERA	33000			$ 4004524
RADIOLOGY	10000			$ 1372798
DIAGNOSTIC RADIOL	5500	90	1.65	816749
~~THERAPEUTIC RADIO~~	1500	90	1.65	222749
SPECIAL PROCEDURE	2000	90	1.65	296999
~~ISOTOPES~~	0	90	1.65	0
RADIOLOGY ADMINIS	1000	22	1.65	36299
LABORATORIES	12000			$ 1737445
ANATOMICAL PATHOL	1000	101	1.65	166649
~~HISTOPATHOLOGY~~	1000	101	1.65	166649
HEMATOLOGY	1000	90	1.65	148499
~~BLOOD BANK~~	500	101	1.65	83324
ISOTOPE LABORATOR	1500	101	1.65	249974
~~BIOCHEMISTRY~~	2000	101	1.65	333299
MICROBIOLOGY	2000	101	1.65	333299
~~LABORATORY ADMINI~~	1000	20	1.65	32999
MORGUE & AUTOPSY	1000	101	1.65	166649
~~ANIMAL QUARTERS~~	1000	34	1.65	56099
~~PHYS MED & REHAB~~	5000			$ 371249
~~PHYSICAL THERAPY~~	2000	45	1.65	148499
OCCUPATIONAL THER	1000	45	1.65	74249
~~RECREATIONAL THER~~	500	45	1.65	37124
BRACE SHOP	300	45	1.65	22274
~~A D L~~	600	45	1.65	44549
VOCATIONAL GUIDAN	600	45	1.65	44549
ELECTRODIAGNOSTIC	2500			$ 280499
ELECTROCARDIOGRAP	1500	68	1.65	168299
~~ELECTROENCEPHALOG~~	500	68	1.65	56099
BASAL METABOLISM	0	68	1.65	0
~~ELECTROMYOGRAPHY~~	500	68	1.65	56099
~~PARAMEDICAL SERVIC~~	1500			$ 94049
~~INHALATION THERAP~~	500	38	1.65	31349
I V THERAPY	500	38	1.65	31349
~~TRACTION SERVICE~~	500	38	1.65	31349
~~MISC ADJUNCTS~~	2000			$ 148499
~~CARDIOPULMONARY~~	1000	45	1.65	74249
RENAL DIALYSIS	1000	45	1.65	74249
~~PSYCHOLOGY SERVIC~~	0	24	1.65	0
EDUCATION	9500			$ 376199
EDUCATION FACILITI	9500			$ 376199
MEDICAL EDUCATION	4000			$ 158399
UNDERGRADUATE	2000	24	1.65	79199
~~GRADUATE~~	1000	24	1.65	39599
POST-GRADUATE	1000	24	1.65	39599
NURSING	2500			$ 98999
IN-SERVICE	1000	24	1.65	39599
~~AFFILIATED PROGRA~~	1000	24	1.65	39599
NURSING SCHOOL	500	24	1.65	19799
MISC EDUCATION	3000			$ 118799
TECHNICAL EDUCATI	500	24	1.65	19799
~~MEDICAL LIBRARY~~	2000	24	1.65	79199
AUDIO-VISUAL TEAC	500	24	1.65	19799
~~RESEARCH~~	56000			$ 3493868
~~RESEARCH FACILITIE~~	56000			$ 3493868
~~RESEARCH SPACES~~	56000			$ 3493868
~~LABORATORY RESEAR~~	35000	38	1.65	2194499
CLINICAL RESEARCH	3500	35	1.65	202124
~~ANIMAL FACILITIES~~	3500	38	1.65	219449
RESEARCH ADMINIST	3500	38	1.65	219449
~~COMMUNITY MEDICAL~~	3500	38	1.65	219449
SOCIAL RESEARCH	3500	38	1.65	219449
~~AFFILIATIONS IN R~~	3500	38	1.65	219449

4





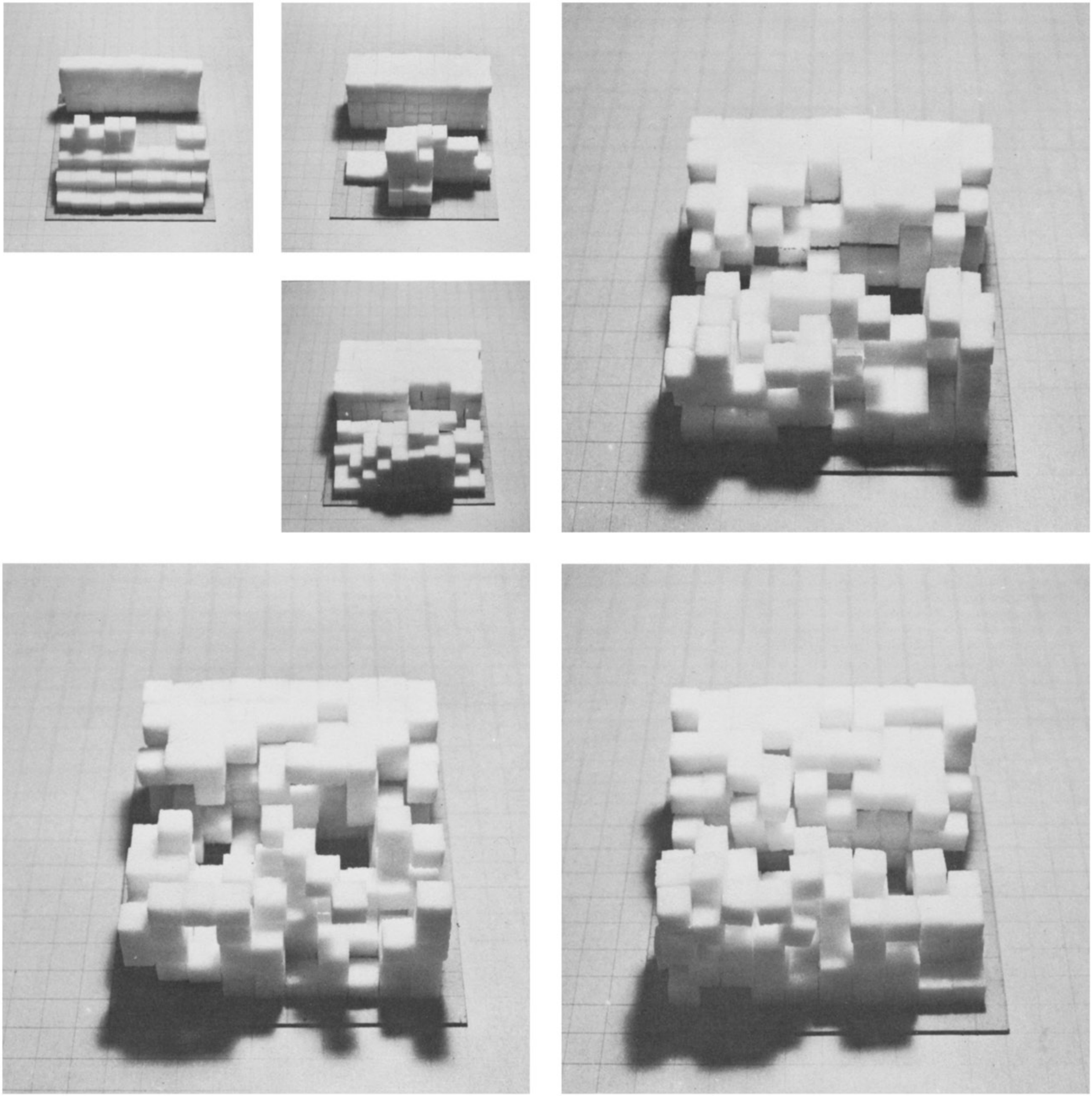


1 GRASP, Generation of Random Access Site Plans. This computer program generates solutions only within an underconstrained situation where the operator specifies dimensions, "no-build areas," density, cost, and aspects of privacy. "Good" solutions can be plotted in perspective or orthographic modes. (Work by Eric Teicholz. Illustration courtesy of the Harvard Laboratory for Computer Graphics)

2 Two outputs of COMPROGRAPH 3 by Eric Teicholz and Thomas Follett for architects Perry, Dean, and Stewart. The user prepares a three-dimensional matrix as input, specifying size and functional relationships. After specifying the envelope, radial or linear, the routine will generate schematic plans on a floor-by-floor basis. This particular computer program promotes a present-day method that in itself is debatable and is certainly questionable in the light of emerging computer techniques. (Illustrations courtesy of the Harvard Laboratory for Computer Graphics)

3 RUMOR, The random generation and evaluation of plans. A matrix of relationships is established by the operator for each criterion. "No effort has been made to generate only 'good' plans" (Bernholz, 1969). The two illustrations represent a house plan composed of a living room, dining room, kitchen, four bedrooms, one bathroom (a debatable functional relationship), a TV room, a washroom, and a sewing/laundry room. (Illustrations courtesy of the Harvard Laboratory for Computer Graphics)

4 A preliminary output from the Children's Hospital Project of the Leo A. Daly Company, Architects. The 134 activities are given minimal interrelationships. While talking with a particular designer, the program implicitly develops functional relationships through trial and error, punishment and reward. Over time the system should improve. It is now under research by Stephen Flanders and Lee Windheim, using the Service Bureau's CALL 360. (Illustrations courtesy of the Leo A. Daly Company)

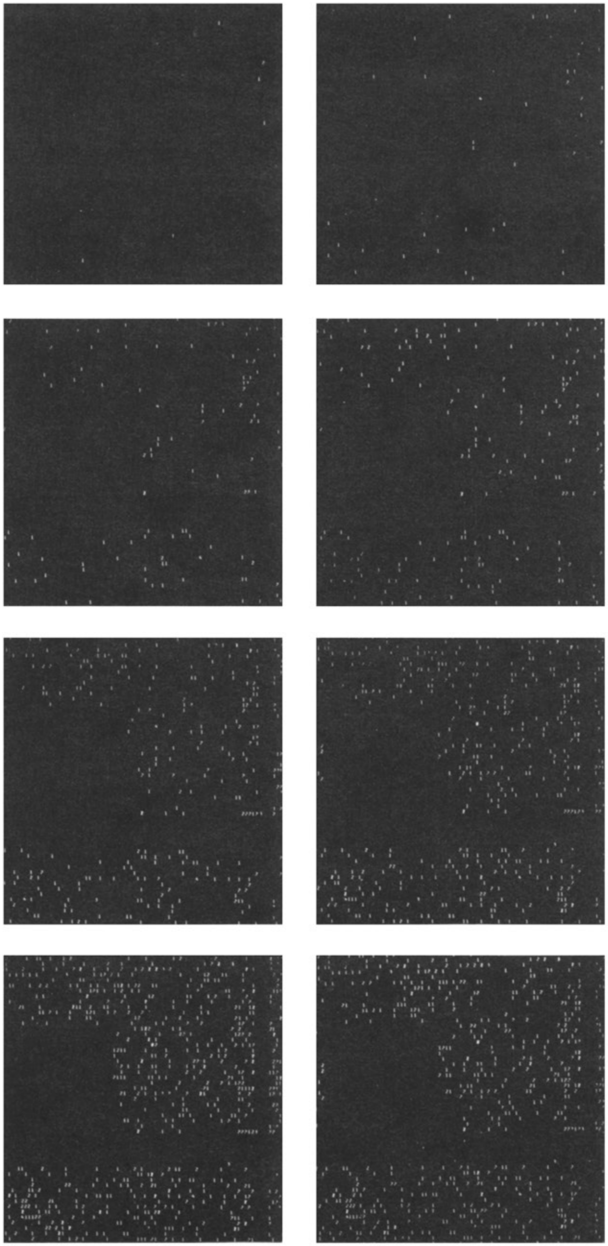


The three small illustrations are models of three of the ten inputs to LEARN. The remaining illustrations are representative of the outputs at different time intervals. The work was performed by Anthony Platt, Peter Bailey, Gary Ridgdill, and William Hurst.

cedures can appropriately acknowledge this sort of growth by changing the distribution of "randomness" in response to the present state of the form, as described by previous actions, external information, and stage of growth.

As one example of solution generation, a student project—LEARN—was developed by a group of M.I.T. Master's of Architecture students who had no previous computer programming experience. LEARN was a computer mannerist. It watched the designers' activities by observing ten simple solutions. (In this case they were "sugar-cube" models transcribed to punch cards describing *x-y-z* centroid locations of solids and voids). Following these ten archetypes, the machine was asked to generate a solution of its own. The appeal of this simple experiment is that the criteria were first determined from the form and then used in the generation of the alternatives. The students observed the variations within the given "style" of the solution. The mannerism was derived from the original ten solutions and was then updated by the eleventh. The machine proceeded to generate a twelfth solution, updated its "manner," generated a thirteenth, and so on. After a denouement of five thousand separate solutions to the same problem, the mannerist machine did not generate or embark on wild tangents. In fact, the conviction of the machine was so intense that the last thousand solutions had little distinguishing variety.

A second example is GROWTH, also a student project. This system operated within a larger work space (approximately a square mile) than LEARN and did not observe a specific designer's methods. The generated solutions were



GROWTH. The final run of this program used two hours of dedicated IBM 360/65 computer time to simulate 266 stages of growth. The experiment was conducted by Judd Knoll, John Maugh, and Chin Pai.

The eight illustrations, from top to bottom, represent the following stages with the associated number of solids:

stage-solids
11-11
26-69
35-103
59-205
131-555
179-801
235-1082
266-1251

periodic glimpses at stages of growth. The computer employed the principle of "influences," where each element's status (solid or void) was determined by its "conviction" (to be what it was or to be what it was not). As soon as a void became solid or a solid became void, ripples of influence would disperse, locally disturbing the convictions of adjacent elements (in proportion to proximity and activity relationships). A solid might become more convinced of its solidity or else an adjacent void might tend toward a state of solidity, being now unconvinced of its status. In effect, the rules of conviction were the generating force. For example, a lone ten-foot cube in the middle of a large field might influence its void neighbors under one set of rules to be less convinced of their voidness and accordingly raise their probability of changing state in the next stage of growth. Meanwhile, another set of rules might make the edge members of a large complex thoroughly convinced voids or thoroughly convinced solids. The same rules might tend to lower the conviction of deeply embedded solids (in order to avail the form of interior open spaces in response to size).

A third example is the ongoing research of Timothy Johnson and Richard Krauss at M.I.T., under NSF contract (T. Johnson et al., 1969). Under the direction of Albert Dietz, this space-allocation work employs sophisticated mathematics and sophisticated graphics to optimize cross-coupled constraints and display the results. The generated solutions are composed of "use-surfaces" and "use-volumes." They are the result of optimization techniques that assume missing information (as opposed to replacing it with random numbers). A generated solution is a function of weighted proximities, orientations (site and exposure), visual access, acoustical access, circulation, and others to be implemented. It is displayed to the user for his consent and rearrangement. Subsequently, the machine regenerates a solution more specified than the last but in the same tenor as the last. Because the machine does not explore divergent tacks, it could channel the unwary user in the wrong direction.

A photographic record of a circulation conflict. In this case the simulation is the real world, the best model but the most expensive. Similar displays will soon be manageable by computers. (Photographs by Tom Payne)

Simulation of Events

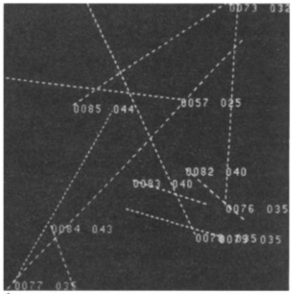
When enough empirical or experimental rules are known about a process, machines can be made to take on the character of the event and undergo a make-believe happening of that process, a simulation. Given reasonable protocols and maxims, this form of mechanical masquerade is a powerful method for refining an original set of rules or pretesting designs and procedures.

The simulation of events can benefit the architect in two ways. If the designer does not fully understand the behavioral aspects of an event, he can play with rules and regulations, searching for recognizable activity patterns. In other words from empirical knowledge of a set of actions and reactions for specific environments, a designer could inductively compose postulates or algorithms applicable in other contexts. For example, if he understands from on-premise measurements the vertical circulation patterns for several different environments, he could describe these environments to the machine and hypothesize seemingly appropriate rules. Then when the machine, using these rules, displays the vertical circulation patterns for the known environments, it reveals the divergencies between the empirical data and the designer's rules—between what he knows he should see and what he does see—giving him information by which to modify the rules, always observing whether the change has a positive or negative effect. Eventually, using a dynamic on-line system, he will be able to converge on rules of simulation that can be applied to other environments.

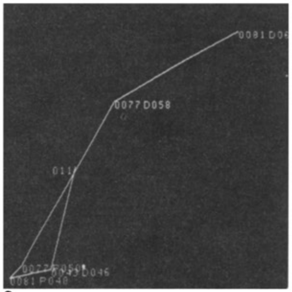
The second design application, pretesting, assumes the rules are correct. Whether empirical or experimental, simulations are no better than their underlying rules, whether the rules are provided by the man or by the machine. If the simulation model is correct, a designer or a machine can observe the performance of an environment, a specific context. Someday, designers will be able to subject their projects to the simulations of an entire day or week or year of such events as use patterns and fast-time changes in activity allocations. On display devices, designers will be able to see the incidence of traffic jams, the occurrence of sprawl, or sweltering inhabitants searching for shade. For the present discussion, the most easily reproducible event is circulation, a perplexing and important urban situation in itself.

Many sophisticated organizations have spent time and money in programs that simulate circulation, primarily vehicular circulation. Rather than observe these elaborate simulation techniques, let us observe two very simple circulation models that have been devised to simulate pedestrian movement. The models result from two M.I.T. student projects involving architecture students, once again with almost no previous programming experience.

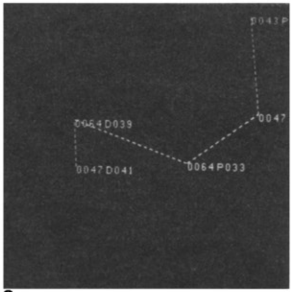
The first simulation model describes three parameters: spaces (function, capacity, desirability), circulation interfaces (direction, capacity, demand) and people (arrivals, departures, frustrations). The model assumes the chosen environment to be a discrete chunk of the real world, with a certain number of pedestrians leaving the system and others arriving at each time interval. At each instant, the circula-



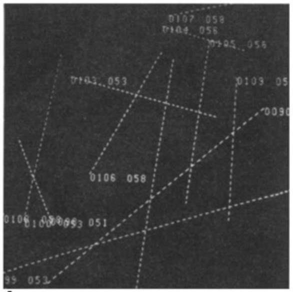
1



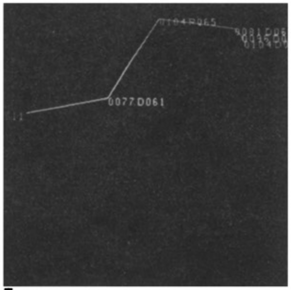
2



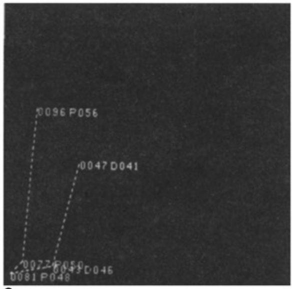
3



4



5



6

CARS — computer-automated routing and scheduling. This system is designed to provide door-to-door transportation in low-density suburban areas. The aim of CARS is to provide service approximating that of the taxicab, but at a price approximating mass transit bus systems.

The six illustrations represent a simulation of CARS in operation with twelve vehicles on an area of nine square miles with about ninety demands per hour. Two particular criteria are enforced: no one should wait more than fifteen minutes, and no one should travel more than 1.8 times the direct driving distance.

The illustrations have been photographed from an ARDS tube which runs off the M.I.T. time-sharing system. The work is being performed at M.I.T's newly created Urban Systems Laboratory under the direction of Daniel Roos.

1 All waiting demands at time 45

2 The projected tour of vehicle 11 at time 45

3 A history of vehicle 11 at time 45

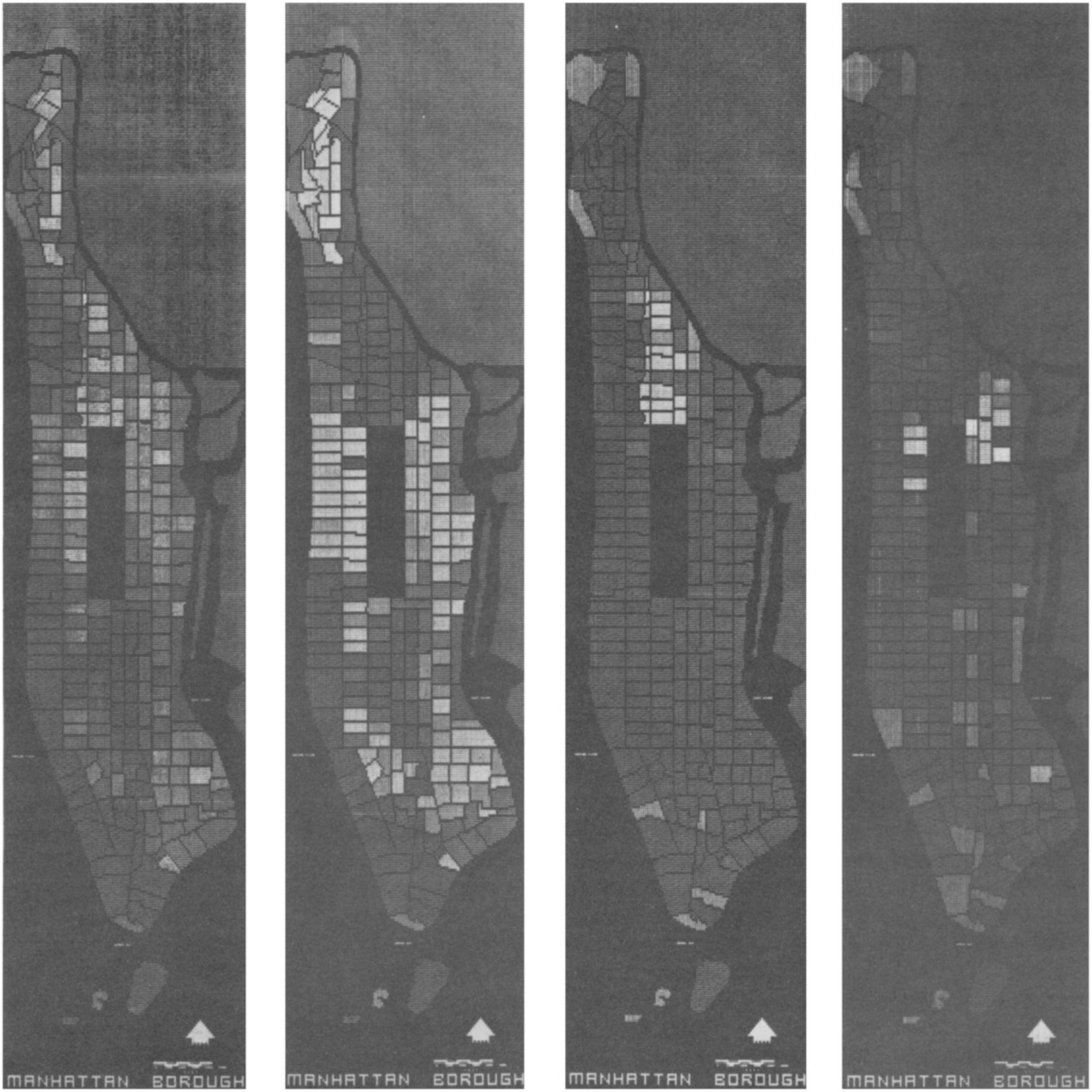
4 All waiting demands at time 60

5 The projected tour of vehicle 11 at time 60

6 A history of vehicle 11 at time 60 tion activity and the space populations are determined by random numbers controlled by parameters of frustration and desire. Although this work was not implemented on a graphic display device, the authors (with some effort) can observe jammed doorways, vacant commercial spaces, and periodic peaks in major circulation routes. Their physical model can be changed and manipulated in search for less antagonistic circulation patterns, iterating toward a design solution that would display ambulatory ease and facility.

In this example, simulation techniques describe agglomerates of people, whole groups moving from space to space in one cycle. The second student model applies variable parameters to each individual pedestrian. Characteristics of desire and destination control the simulated movement of each individual pedestrian in accordance with his local environment. The student can observe frustrations and localized frictions that are not only a function of the physical form but responses to the individual personalities of the other pedestrians in the same space. The student can observe a dashing blonde unsettle corridors or a precipitate fleet-footed latecomer disrupt a reception area.

Both student projects, even in their infancy, exhibit viable methods for prediction. When simulation techniques improve and are part of architecture machines, physical structures can be tested within environments that acknowledge their presence. In other words, when a change is contemplated for some neighborhood, it can be tested by observing its effect over time, but in fast time, unreal time. Usually, in the nonpretend world, the real world, a neighborhood immediately responds to a change, generates new demands, and the supposedly beneficial event is too often invalidated. Such negations can be avoided. Direct interplay between event and effect, desire and result can be observed and can be enveloped in simulation procedures.


MANHATTAN BOROUGH
MANHATTAN BOROUGH
MANHATTAN BOROUGH
MANHATTAN BOROUGH

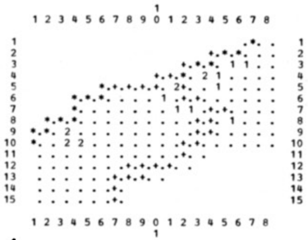
SYMAP. This computer system is primarily concerned with the display of spatially distributed data rather than with its manipulation. Developed by Howard Fisher, it employs an overprinting technique on a high-speed printer, which in the days before computer graphics was fine, but is quite obsolete today. The four maps are based on the 1960 census and are at the scale of the census tracks. From left to right, they display density per acre of total population, whites, blacks, Puerto Ricans. The work was performed by Peter Rogers and Isao Oishi. (Illustrations courtesy of the Harvard Laboratory for Computer Graphics)

Bits of Design Information

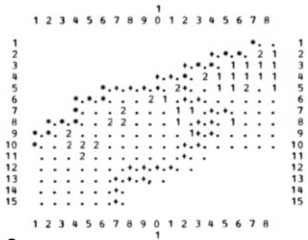
Census data, site descriptions, transportation statistics, activity constraints, economic criteria, and material specifications are all part of the bulky dossier of design information necessary for any urban design project. The information burden is fantastic. What usually happens in most design procedures is that a handful of criteria are chosen and thoroughly developed; all the remaining information relationships are expected to fall into place, or else residual issues are crammed into unsuspecting receptacles. Or in a gesture of design fatalism, accepting the unfeasibility of it all, a group of parallelepipeds are contrived and refined to accommodate as well as possible the internal demands of some institution. The problem and the result are commonplace—look at your own city.

An architect's role in urban design requires a complex information supply with characteristics of retrieval, labeling, and interassociation. But machines are good at this. Though there are technical problems and real computer-programming issues, machines *can* respond to and have access to millions of billions of bits of information. It is estimated (Servan-Schreiber, 1967) that the number of all letters in all words in all books in all libraries in the world exceeds one thousand trillion (1,000,000,000,000,000). J. W. Senders (1963) estimates that the current growth rate of this store is about four hundred thousand letters per second. Even a modest architect might assume that he needs some of this store.

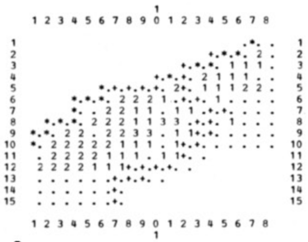
In the human nervous system, information genuinely constitutes authority (McCulloch, 1965). In



1



2



3

4

DISCOURSE

In the five maps, the following symbols apply:

1 = residence
2 = industry
3 = centers
B = residence and centers
C = industry and centers
+ = river
− = channel
* = channel and river

1 Ciudad Guayana 1961, showing residence and industry

2 Ciudad Guayana 1969, showing residence and industry

3 The proposed design for Ciudad Guayana generated by designers from the Venezuelan Development Corporation for the Guayana Region

4 Two patterns generated by DISCOURSE decision rules (Illustrations courtesy of William Porter) design, however, abundant data can confer prestige on mediocre designs, especially when facts arrive from the unequivocating computer. Data can be prepared to support any design if the selection of evidence is limited to that which favors the cause. "Poor data and good reasoning give poor results. Good data and poor reasoning give poor results, poor data and poor reasoning give rotten results" (Edmund Berkeley, 1967).

A machine could store relevant information in many ways. Relational and associative data structures, for example, store classes of items by properties of similarity and retrieve them by querying for that which has "this and that, but not those." Another structure uses lists of attributes that point "fingers" at members that have the same attributes, thus tying threads among the various members of the data structure. Still another (and simpler) method is a matrix organization where rows and columns of entries are entered and looked up by addressing a two-, three-, four-, or *n*-dimensional table.

But in architecture, most information has a natural disposition—the positional relationship—which can help to organize the proliferation of data. Design manipulations invariably wield locational data expressed in terms of position, distance, area, or volume. This natural geometrical referencing suggests a data structure where each physical location (solid or void, building or open space), to as small a grain as possible, would describe itself in an autonomous fashion (even the voids!). This has strong implications, especially the Euclidean and redundant nature of geometrically related data.

Information search, by either designer or machine, would occur for the most part in a localized fashion, investigating by proximity (by neighborhood, by street, or by immediate adjacency). The thrust of this sort of data-structure argument is that information is treated locally, by positions, and less globally, by attributes. Thus, design information is retrieved by geometric (topical) search rather than by intersecting generalities.

Such a position-oriented storage vehicle may be unique in the physical design problems of the urban environment. In a library reference system, this type of information structure is ridiculous; books are not categorized by their position on the shelf, books are redundantly classified by name, author, subject, publisher, and so forth. Unfortunately, good library data structures are all too often foisted onto design problems.

One particular design information system—DISCOURSE—warrants mention, as it exemplifies a flexible data structure that combines the assets of associative and matrix organizations, attribute and geometric searches. This research team (Fleischer et al., 1969, and Porter et al., 1969) uses the M.I.T. time-sharing facilities to interact (no dialogue) with data files and print the results in tabular or map format. It is a problem-oriented language that derives flexibility from (1) providing multiple data structures for both local and global interrogation, and (2) providing a "meta-language" that allows the designer to create his own search techniques. The reader must understand, however, that DISCOURSE is not computer-aided design within *our* definition. It is an excellent computer system that manipulates bits of design information, that is, information that has been explicitly given to the machine by the user.

Another example is MEMORY, an information storage and retrieval system that is being studied within M.I.T.'s Urban Systems Laboratory. MEMORY's dominant feature is its "forgetting convenience." It is a way of storing events in neural nets that are highly redundant and, at first, rather random. Over time and through repetition or the lack of it, events become, by the strength of traces left in memory, either stronger remembrances or fainter recollections. At the onset of such a system, for any given input the output will be mostly garbage. Over time, the responses should gain meaning with respect to both the input and the relevancy (defined by time) of the input. The reason that this experimental work is important to an architecture machine is that the design process is an evolution of (1) the product, the form; (2) the process, the algorithms; (3) the criteria, the information. MEMORY addresses itself to item number three.

Machines in Residence

Modern decision theory, economics, psychology, and game theory recognize, as a basic case, clearly motivated individual choice under conditions of complete information. It is also recognized that two unfortunate facts of life remove us from the relative simplicity of this basic case. The first concerns man as an information processor and the second the conflict of individual with group preferences.
Martin Shubik, "Information, Rationality and Free Choice in a Future Democratic Society"

Lower-class people need big kitchens; middle-class people need big bedrooms; corridors are for the poor, and so forth. Design universals enable federal housing authorities to set minimum standards, they enable architects to disregard specifics, they delight lovers of empirical generalizations. In short, empirical generalizations of life styles are for the comfort and convenience of the decision makers' tools, not necessarily for the well-being of the people.

Today we have "advocacy planning," a design procedure that tries to overcome the lumping of life styles, that tries to satisfy particular requirements. Attempts to procure individual needs and desires have embodied several formats: the questionnaire (fill in the missing spaces), the neighborhood meeting (we are here to listen to your problems), the personal interview (tell me what you want). Note that in each of these communications media it is assumed that the asker knows what to ask, the answerer knows what to answer, and that minds will not change rapidly. Furthermore, advocacy planning is conducted in such unreal time that the fancies of the individual householder change in the lapse of time.

Before suggesting procedures that are more appropriate to the articulation and satisfaction of local desires, let us first assume two future technological advances: versatile building systems capable of responding to changing (per month, season, year) human needs and the direct concern of this book, home computer terminals capable of talking in a graphic and auditory fashion—"but I don't see any computers getting into my house" (A. Milne, 1963).

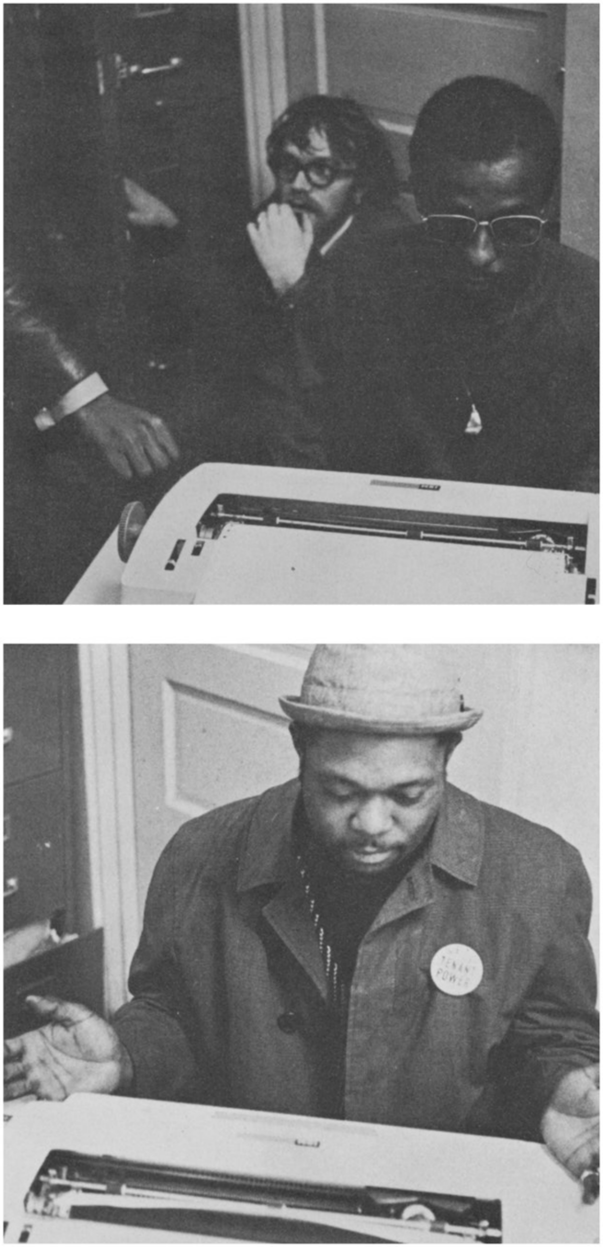
You need not look too far, maybe ten years: ". . . computer consoles installed in every home . . . everybody will have access to the Library of Congress . . . the system will shut the windows when it rains" (McCarthy, 1966). Such omnipresent machines, through cable television (potentially a two-way device), or through picture phones, could act as twenty-four-hour social workers that would be available to ask when asked, receive when given. Imminent changes in family size could be overlaid upon a local habitat in an effort to pursue growth that would not curtail the amenities children need.

Granting machines in the home, each urbanite could intimately involve himself with the design of his own physical environment by (in effect) conversing with his own needs. Or, another way of thinking of the interaction is that everybody would be talking to the architect, not explicitly but implicitly, via a machine-to-machine interchange. Architects would respond to particular patterns of a neighborhood and submit alternatives to be played with and in such a manner possibly penetrate the designer-dweller dissonance that exists in today's housing problem.

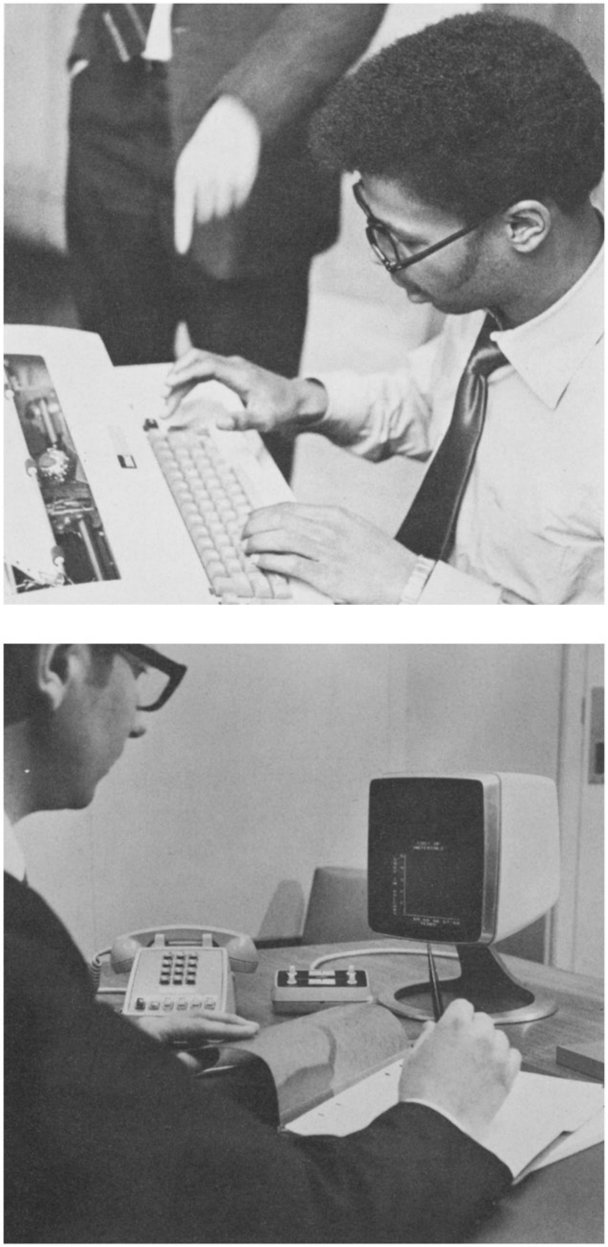
Even today, the touch-tone telephone gives rise to a home computer terminal whose ten-button dialect humors a potentially ubiquitous man-machine conversation. Coupled with audio response units, such telephones can converse with button-pushing as an input and spoken English as an output. Frank Westervelt (and Smith, 1968) has incorporated such a system at the University of Michigan's Computation Center.

Richard Hessdorfer is expanding Westervelt's system by constructing a machine conversationalist. Hessdorfer's work is aimed at initiating conversation with an English-speaking user. His problem is primarily linguistic. The machine tries to build a model of the user's English and through this model build another model, one of his needs and desires. It is a consumer item (as opposed to an industrial or professional tool) that might someday be able to talk to citizens via touch-tone picture phone, or interactive cable television.

As a part of the Hessdorfer experiment, a teletypewriting device was brought into the South End, Boston's ghetto area. Three inhabitants of the neighborhood were asked to converse with this machine about their local environment. Though the conversation was hampered by the necessity of typing English sentences, the chat was smooth enough to reveal two important results. First, the three residents had no qualms or suspicions about talking with a machine in English, about personal desires; they did not type uncalled-for remarks; instead, they im-



1



2

1 The three protagonists of the Hessdorfer experiment, Maurice Jones (top right), Barry Adams (top left), and Robert Quarles (bottom left). It is interesting to note the button Robert Quarles happened to be wearing that day: "Tenant Power."

2 Picturephone. Copyright 1969 Bell Telephone, Inc., Murray Hill, New Jersey. Reprinted by permission of the Editor, Bell Laboratory RECORD.

mediately entered a discourse about slum landlords, highways, schools, and the like. Second, the three user-inhabitants said things to this machine they would probably not have said to another human, particularly a white planner or politician: to them the machine was not black, was not white, and surely had no prejudices. (The reader should know, as the three users did not, that this experiment was conducted over telephone lines with teletypes, with a human at the other end, not a machine. The same experiment will be rerun shortly, this time with a machine at the other end of the telephone line.)

With these domestic (domesticated) machines, the design task becomes one of blending the preferences of the individual with those of the group. Machines would monitor the propensity for change of the body politic. Large central processors, parent machines of some sort, could interpolate and extrapolate the local commonalities by overviewing a large population of "consumer machines."

What will remove these machines from a "Brave New World" is that they will be able to (and must) search for the exception (in desire or need), the one in a million. In other words, when the generalization matches the local desire, our omnipresent machines will not be excited. It is when the particular varies from the group preferences that our machine will react, not to thwart it but to service it.

011

Aspects of Design Processes

Sequential and Temporal Events

A process is a progressive course, a series of procedures. A procedure is replicable (if you understand it) in an algorithm; its parts have a chronological cause-and-effect relationship that can be anticipated. A procedure can be replicated with the appropriate combination of commands. In short, a procedure is deterministic and can be computerized within a given context.

Conversely, a process cannot be computerized, but, as we have said, it can be computer-aided. Particularly in the design process, respective events are not chronologically ordered. The following scenario, without the enrichment of graphics, intonations, bodily involvement, crudely illustrates an architect-machine dialogue:

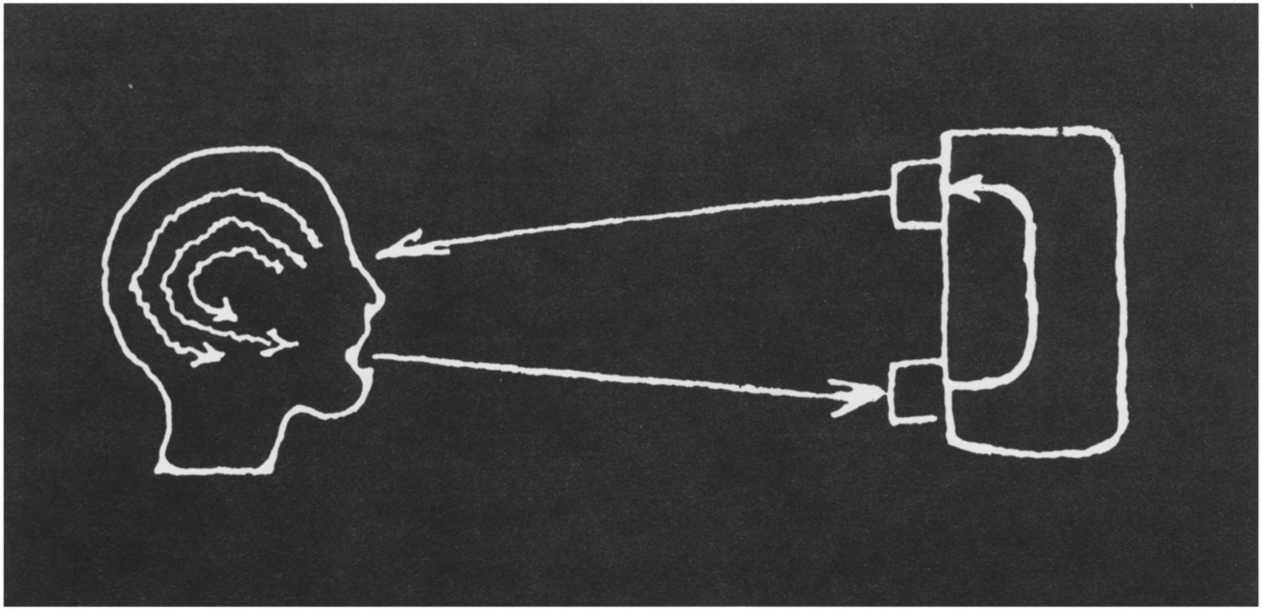
Machine:
George, what do you think about the children's activities in this project?
Architect:
How far must a child walk to nursery school?
Machine:
The average distance is 310 feet.
Architect:
Each dwelling unit must have direct outdoor access and at least three hours of direct sunlight.
Machine:
Of the children we were just discussing, 92 percent must cross a road to get to school.
Architect:
We will look at that later. With respect to dwelling units, we must assume at least two vehicles per family.
Machine:
Your ozalids are ready. Your wife has just called. . . .

The example describes a participation where each party is interjecting and superpositioning events directed toward a common goal.

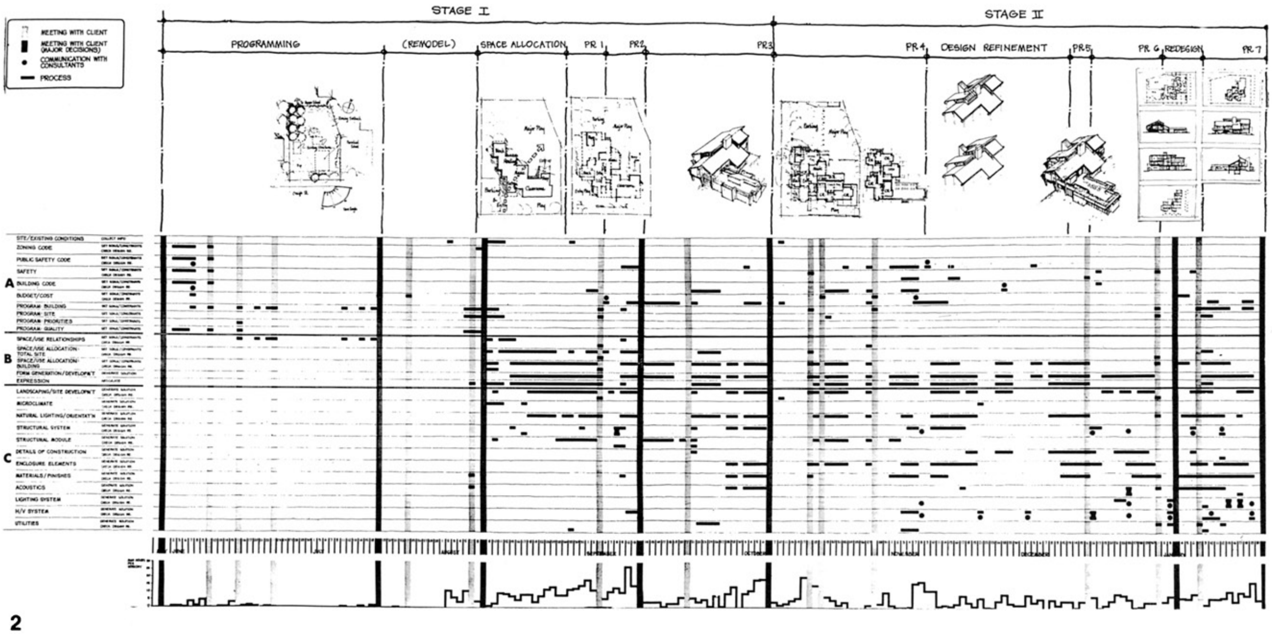
Each event is either a temporal or sequential occurrence; together they constitute part of a process. A sequential response of one protagonist is generated by the previous event in the dialogue, usually on the behalf of the other. A sequential event is a reply. It can be the reply to a facial expression or the answer to a question. What is important, however, is that not only is one actor responding but he can assume that the other is listening and probably is aware of the context. In other words, a sequential episode assumes the reply of one (intelligent) system and the attention of the other system—a chain of chronologically ordered incidents.

This well-known command-and-reply relationship between man and machine does not in itself constitute a dialogue, as it ignores all events except those ordered by time sequence. The Soviet Union's A. P. Yershov (1965) has a diagram illustrating this proverbial man-machine interaction, as he calls it, "director-agent" interaction. Note that in the diagram, Professor Yershov has drawn three arrows within the man's head and only one arrow within the machine. The three arrows imply an ever-continuing act particular to the role or constitution of the man and not the machine. Let us call this act deliberation.

The act of perpetual cogitation can be equally accorded to machines, especially since we have



1


MEETING WITH CLIENT
MEETING WITH CLIENT (MAJOR DECISIONS)
COMMUNICATION WITH CONSULTANTS
PROCESS
STAGE I
STAGE II
PROGRAMMING
(REMODEL)
SPACE ALLOCATION
PR 1
PR2
PR3
PR4
DESIGN REFINEMENT
PR5
PR 6
REDESIGN
PR 7
SITE/EXISTING CONDITIONS
ZONING CODE
PUBLIC SAFETY CODE
SAFETY
A
BUILDING CODE
BUDGET/COST
PROGRAM: BUILDING
PROGRAM: SITE
PROGRAM: PRIORITIES
PROGRAM: QUALITY
SPACE/USE RELATIONSHIPS
SPACE/USE ALLOCATION: TOTAL SITE
B
SPACE/USE ALLOCATION: BUILDING
FORM GENERATION/DEVELOPM'T
EXPRESSION
LANDSCAPING/SITE DEVELOPM'T
MICROCLIMATE
NATURAL LIGHTING/ORIENTAT'N
STRUCTURAL SYSTEM
STRUCTURAL MODULE
DETAILS OF CONSTRUCTION
C
ENCLOSURE ELEMENTS
MATERIALS/FINISHES
ACOUSTICS
LIGHTING SYSTEM
H/V SYSTEM
UTILITIES

2

1 The Yershov diagram. This first appeared in a paper presented at a Seminar on Automation of the Thinking Process held at the Kiev Center for Scientific and Technical Information, Kiev, USSR, 1963.

2 Design History Chart (Myer and Krauss, 1967). The diagram represents a series of procedures rather than a process. The chronology of left to right, the two-dimensional apsects of the printed page, and the start/finish overtones all contradict the nature of the design process. If it were possible to diagram this process, then such diagramming would occur out of context, and present-day machines could handle it without an artificial intelligence. (Diagram courtesy of the M.I.T. Center for Building Research)

previously insisted on a dedicated small machine in residence, devoting its full computational ability full time. We will call machine deliberation "temporal" work. It resides in the background and surfaces as an interrupt. The interrupt (though not necessarily the deliberation) is context-dependent; thus we can probably assume that the temporal zone requires an intelligence. Furthermore, note that it is this zone of temporal events that the designer interrupts when presenting a fact or a task.

In the foregoing sketch, the machine addresses the architect, presumably interrupts him. Following, the architect addresses the machine (in fact interrupts) with a specific question that is not a reply but is within the same context. The machine's reply is sequential: ". . . 310 feet." While the architect thinks about the response, the machine further investigates the children-nursery relationship (we assume here a previous experience by the machine with such issues). Within three seconds of user deliberation, a machine could devote between three hundred thousand and three million operations to the children-nursery relationship.

Meanwhile, during the machine's activities, the architect reinterrupts the machine and states criteria with reference to a new context: "Each dwelling unit must have direct outdoor access and at least three hours of direct sunlight." After the machine has listened (and heard), it interrupts the architect and lets surface from the temporal zone the unsolicited information about children's circulation. The architect postpones consideration of his oversight and proceeds to supply further design constraints.

Following, the machine interrupts again with a time-dependent occurrence.

It may now be more evident why an evolutionary machine must have the capacity for context recognition. A complete mishmash of irrelevant comments from a nonintelligent, nonevolutionary machine would confuse the designer and thus stifle the design process. While at the onset of any partnership the machine's interruptions might appear random or disorderly, they would gain relevance through evolution. The sophistication of these temporal actions is essential for machines to mature into intelligent partners.

The Geometry of Qualities

In order to make adept, temporal comments, an architecture machine must have a certain basic understanding of qualities. Though at first primitive, this qualitative appreciation itself would evolve within a value system that is very personal, between a man and a machine.

The handling of qualitative information is too often presumed hopeless for the constitution of machines. Or it is granted feasibility only through the abortive techniques of quantification. No doubt, characteristics of identity, oppression, and fulfillment are hard for our present machines to comprehend. Nevertheless, even with existing machines, properties of privacy or accessibility or the natural environment furnish qualitative features that can be readily expressed in terms that are understandable to machines, machines that for the time being have not experienced these qualities. This is because we already have a model whose base is geometry. This geometric structure, resulting from the form base of urban design and the pictorial structure of computer graphics, happens to suit the topology of many environmental qualities.

For example, within some context, visual privacy has an explicit geometry. The presence of a transparent surface, while providing light and view, might not yield visual privacy. A machine can check this without disturbing the architect, by weighing the activity (sleeping, eating, bathing), the actor (housewife, bachelor, exhibitionist), the external uses (commercial, recreational, residential), and the geometry of the two spaces that abut the surface. With an evolutionary designer-machine agreement of the definition of visual privacy, the four ingredients can determine either an unequivocal absence or presence of visual privacy, or a graded value of it.

Unfortunately, visual privacy has psychological and personal ramifications not expressible in the four parameters. These subjective and personal parameters are important; however, they are more appropriately manipulated by the inhabitant (and his machine) rather than the designer. A prospective lessor or buyer, in conversation with his terminal (less elaborate than an architecture machine), can placate his need for privacy by manipulating surfaces and volumes in a given framework. Thus we have a situation where a general scaffolding is locally nourished by residents managing their own insular needs. The concept of an architect (a professional) handling topical qualities and each urbanite interjecting personal standards is particularly compatible with the notion of "plug-in" environments. Machines are the architects for a range of qualities (using human or nonhuman values) that structure the environment, the architects are architects for a piano nobile of qualities, and the householders are architects for local qualities.

As Peter Cook (1967) asks, "Does consumer choice of pre-fabricated living units and the like imply that every man might become his own architect?" Or, as another author suggests, "The housing modules can be bought and sold much like cars, new or second hand. . . . The individual units can be combined vertically and horizontally . . . residents will buy and own their housing modules, but rent the space they occupy" (Hosken, 1968).

An architect attempting to provide natural amenities, a resident trying to overlay his own needs, and a machine endeavoring to transcribe these qualities through some geometry all together comprise a system that must always be in equilibrium. The maintenance of this equilibrium is the design process. Within this definition, the urban environment is a multitude of quantitative and qualitative, local and global, individual and group forces that push and pull on a membrane. The shape of this adaptable membrane at any instant of time is urban form.

In effect, the graphic manipulations from many remote terminals would manipulate the urban form. Each action, by designer, by resident, or by satellite machine, would generate repercussions throughout the system. In most cases, effects of a change would have local impact and lose force within several hundred feet of the modification. Effects of a highway or the equivalent of the year 2000 would have more global effects than a family adding a bedroom. But, given a machine that can interpolate qualities, design by-products would no longer be unforeseen civil disasters.

About Unsolicited Notes and Comments

You never actually told the machine that you were interested in lepidoptera, but the machine is finding out—from experience. It contains, that is, a "learning model" which stores, measures, sorts and computes the probabilities of your interests, reactions and ways of thinking. It is learning about you all right, and will soon be giving you extra information about butterflies.
Stafford Beer, "Cybernetic Thrills and Threats"

For a machine to present uninvited comments upon the qualities of a design may seem presumptuous. Yet consider that these observations might well fall into the category of "If I had only thought of . . . ," and so forth. Furthermore, in an evolutionary system any continual and machine-initiated surveillance would be guided by a joint maturing of the architect's ideas along with machine observation of his methods, problems, and intents.

You are designing a soap tray, for example. Sitting at your graphic terminal with your machine, you draw an open rectangular box and specify that it is to be formed from a continuous sheet of moldable plastic. All of a sudden a bell rings or a voice speaks or some text appears on the television screen, bringing to your attention the lack of any drainage facility. How did the machine know enough to make the observation?

There are three sources for such unsolicited comments. First, you could previously have stated very specifically that all soap trays must drain water. The criterion is specific. The machine implicitly applies this maxim to its observation of your soap-tray. In this case the machine's notice is simple and unsolicited only in time, not content.

A second way, at the other extreme of complexity, is through direct experience and real-world observation. For example, a robot might have seen bathrooms, observed soap being used, or fumbled with soap trays on its own. Such a machine might witness soap melting in water and from that make the necessary chain of observations to assume that . . . and so on. Even though this type of learning exceeds the scope in time of our interests, it is important that learning through groping not be underplayed or ignored; in the distant future that is how machines will probably do their learning.

A third method, more realizable in the near future, is through deduction. For example, in describing the function and the environment of the soap dish, you might have stated that soap melts in water and water runs downward. The machine, with the knowledge of the tray's geometry and the surrounding activities, could deduce that water would indeed collect in the same place as the soap. And, since soap melts, a conflict would exist; either the soap or the water must go elsewhere.

Such machine scrutiny is particularly interesting. The facts used to deduce that the collection of water was a conflict are not necessarily unique to the design of soap trays. Water collection is a problem with roofs, sills, pavements, and so forth. In other words, after a few years of evolutionary dialoguing, a designer and a machine can establish a large repertoire of low-level axioms from which the machine can temporally deduce high-level conflicts.

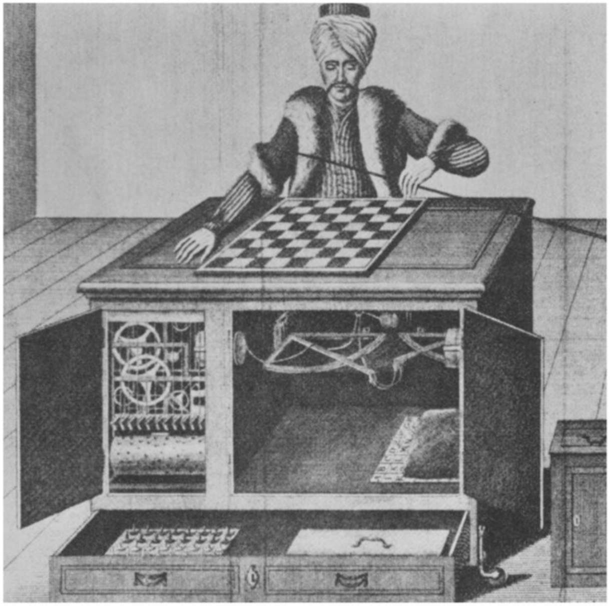
But now the question arises: Why must each architect struggle with indisputable facts? He should not. Simple events—water runs downward, the sun rises in the east—would be built into the machine's design pedigree. Their combination and association, however, must be unformed at the onset and must mature through deducing conflicts in the course of a partnership. In other words, a built-in knowledge may exist that, for example, children do not always look where they are going, and cars can kill. However, the constraint that children must not cross roads alone to get to nursery school would not be an embedded maxim.

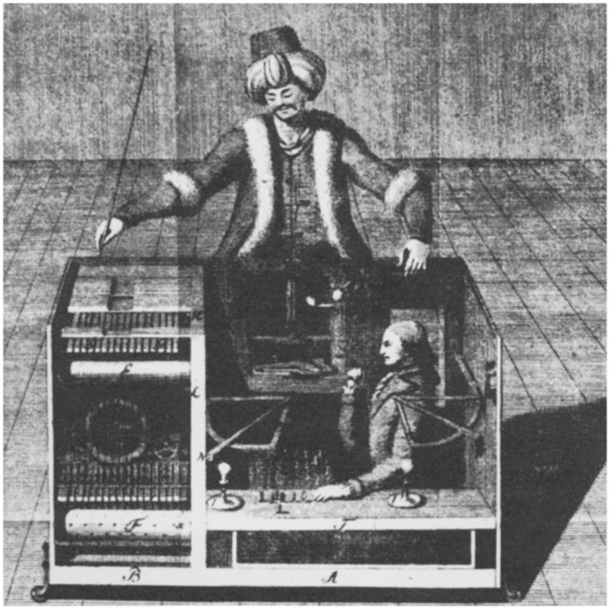
Given a set of axioms and a set of deductive procedures, how does a machine establish the timeliness of an observation? Through context. Three types of context are particularly important: an activity context, a time context, and a rate context. Each involves ubiquitous monitoring and observing. We must assume that the machine continually tracks what the designer has been and is doing.

An activity context is the easiest to implement. Here the machine must balance between commenting on apples when the designer is working with pears. Only when the circulation pattern has been ignored by the heating system, for example, would the machine comment, directing the designer's attention from a context of environmental to circulatory problems.

A time context is a chronology of events, a chronology of design development and design procedures. For example, the level of detail is time-contextual. A comment on bending moments is probably inopportune at the early stages of design. Similarly, in another time context, it might be more appropriate for the machine to withhold a disastrous conflict until after a weekend.

A rate context is a fine-grain time scale. It may be the most important of the three. Observation and recognition of work rates could attempt to rhythmize the dialogue. The machine would try to enter a time phasing personal to and compatible with the designer. Some people, in moments of deliberation, might enjoy a barrage of compliments and comments; others might demand complete silence. Moreover, this attitude may change with mood. Machines must discern such moods. A temporal, unsolicited comment, deduced and timely, could be antagonistic. Such prodding, however, dispels complacency and begins to transform machine servants into machine partners.





Baron von Kempelen's Chess Player, sometimes called "Maelzel's Chess player," after its third owner. The top picture comes from a pamphlet published in 1783 by Chrétien de Mechel whose preface included, "The most daring idea that a mechanician has ever ventured to conceive was that of a machine which would imitate, in some way more than the face and movement, the master work of Creation. Von Kempelen has not only had the idea, but he has carried it out and his chessplayer is, indisputably, the most astonishing automaton that has ever existed." (Chapuis and Droz, 1958)The bottom picture was published much later. It shows the accomplice hidden within the automaton. (Photographs courtesy of Editions du Griffon, Neuchâtel, Switzerland)

Games: Local Moves and Global Goals

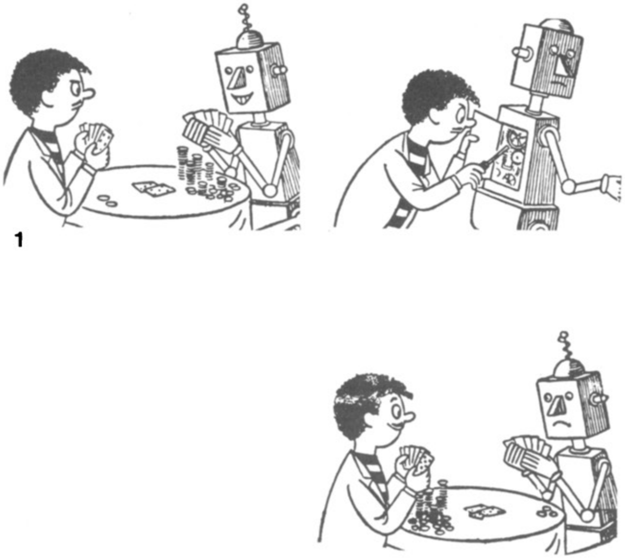
Games provide a happy vehicle for studying methods of simulating certain aspects of intellectual behavior; happy because they are fun, and happy because they reduce the problem to one of manageable proportions.
Arthur L. Samuel, "Programming Computers to Play Games"

Games have fixed rules; gaming involves deception; gamers have opponents. The general game fabric, therefore, is not necessarily consonant with design. Architecture is not Monopoly, Parcheesi, or checkers. Such games assume perfect information, winning is explicit, and the process is composed purely of sequential acts—moves—governed by immutable, fathomable, and predefined rules. Design does not have a clear-cut format; so why is "design gaming" considered avant garde and fashionable? What good are games?

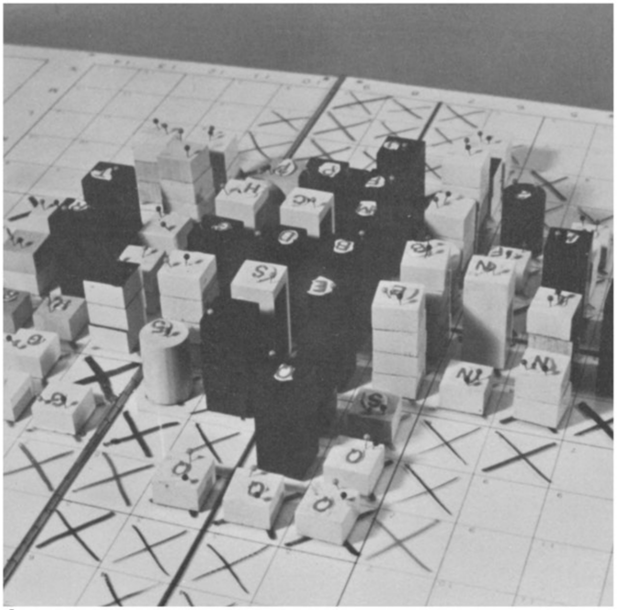
Games are a learning device for both people and machines. "Play and learning are intimately intertwined, and it is not too difficult to demonstrate a relationship between intelligence and play" (McLuhan, 1965). Games are models by which or with which learning takes place. They eliminate worrisome complications and perplexities by using artificially contrived situations. They involve the amalgamation of strategies, tactics, and goal-seeking, processes that are useful outside of the abstraction of gaming, certainly in design.

Historically, chess has been the machine's baccalaureate. In 1769, Baron Kempelen constructed a fraudulent chess-playing machine, The Maelzel Automaton. The hoax was achieved by the labors of a concealed dwarf who observed the moves from beneath and manipulated a mechanical dummy. The need for such fraudulence has since been overcome with computing machinery. The pioneering works of Claude Shannon (1956) and the later efforts of Herbert Simon (and Baylor, 1966) and his colleagues have led to the development of chess-playing machines that demonstrate sophisticated techniques for intelligent decision making by strategically looking ahead. The approximately 1,000,000,000,000,000,000,000,000,000,000,000,000,000,000,000,000,000,000,000,000,000,000,000,000,000,000,000,000,000,000,000,000,000,000,000,000,000,000,000,000,000,000,000 possible chess positions render it improbable that a calculating device can exhaustively search all possible courses of action. As a result, a chess-playing machine looks at local situations, looks ahead some small number of moves, and makes a speculation. Such techniques are indeed relevant to the construction of an architecture machine.

However, rather than map intelligent chess techniques into design tactics, let us concentrate on one key issue in gaming that is particularly relevant to design machines, that is, the relation of local actions to global intents. In architecture the local moves are embodied in physical construction and destruction procedures (whether explicitly executed by a designer or implicitly by zoning laws or the like), and the global goal is quite simply "the good life." In chess, the consensus is that the global goal to win, by taking the opponent's king, has little bearing on the local actions and the skillfulness of making these moves, particularly in



1



2

1 Cartoon that appeared in the Manchester Evening News on May 10, 1957. (Courtesy of North News Ltd., Manchester, England, copyright Copenhagen, Denmark)

2 CLUG, Cornell Land Use Game. CLUG is a game to help humans learn about planning (Wolin, 1968). Each player starts with a fixed amount of cash. The game board is gridded with secondary roads, utility plants are marked, and topographic features can be added. Players risk such real-world disappointments as depreciation, uncontrollable disaster, transportation costs, and so on. The computer in this case, however, is used only as a bookkeeper, keeping participants from losing their interest and making the game move faster when highly paid researchers or officials are playing. (Photograph courtesy of Alan Feldt, developer of CLUG) the opening and middle game. The loser can indeed have played the better game.

In architecture, the losers are rarely the players. This is historically true, but let us assume that it changes and each resident can play the game with the global goal being the good life. The rules for achieving this goal are certainly unclear; they vary for each person, and, as in our *Alice in Wonderland* croquet game, they are ever-changing. Furthermore, in this game there is no coup de grace or checkmate; the global goal has no "utility function," no cost-effectiveness, no parameters to optimize.

But the chess analogy suggests that a machine could learn to play architecture from local design pursuits and that these actions would be draftable without an absolute definition of the good life. A machine's adroitness in design could evolve from local strategies that would self-improve by the machine testing for local successes and failures. In other words, we are suggesting that a machine, as well as any student of architecture, can learn about design by sampling the environment for cheers and boos. For example, in a tennis match a human spectator who is ignorant of the rules, scoring procedures, or criteria to win can begin to distinguish good from bad play merely by observing the applause of the other spectators.

Such learning by inference can apply to the breeding of intelligent design partners able to discriminate between plausible patterns and dubious forms. With a history of local punishments and rewards, an adaptable machine can evolve without a global set of values and adaptable rules to achieve them. Maybe nobody knows how to play, maybe everybody applauds at the wrong time, and maybe the good life is the wrong goal. But the thrust of the game analogy is that we do not have to answer these questions in order to proceed.

URBAN5's Abstractions

In an ideal situation, the communication language could be so informal, that is so natural, that the computer aided designer would not have to learn it. . . . If an incompatibility is found, the designer concerned would be informed. . . .
I. H. Gould, "Some Limitations of Computer Aided Design"

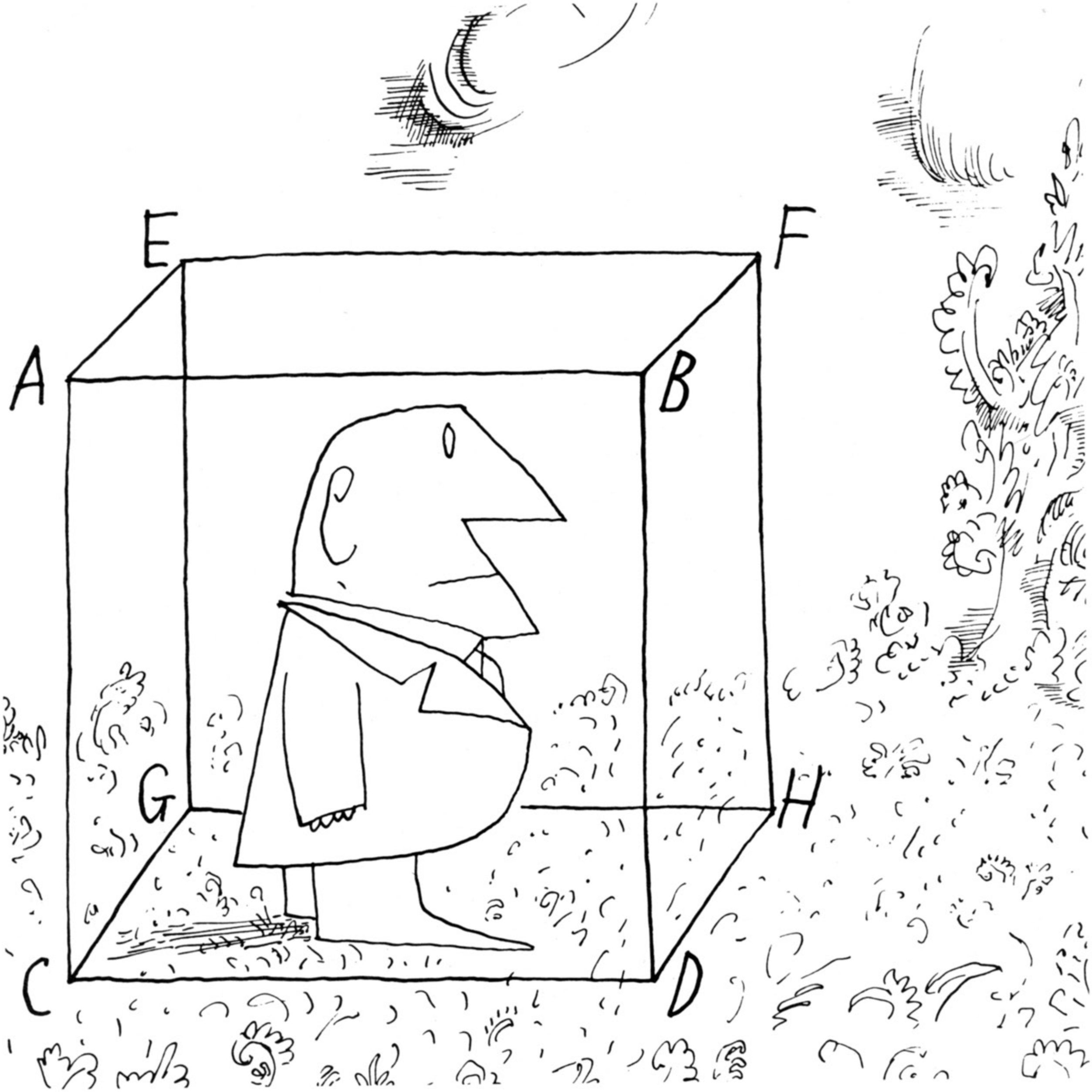
Up to this point, suppositions have been a posteriori reflections upon experiences with the development of the computer system URBAN5. Therefore, this chapter primarily exemplifies some of the previous issues and describes the sequence of events that led to them.

URBAN5's original goal was naively simple. It was to "study the desirability and feasibility of conversing with a machine about an environmental design project . . . using the computer as an objective mirror of the user's own design criteria and form decisions; reflecting responses formed from a larger information base than the user's personal experience" (Negroponte and Groisser, 1967a). The object was to develop a system that could monitor design procedures, in effect, be an urban design clerk.

At the onset of the experiment four assumptions were made: (1) the user is an architect; (2) urban design is based on physical form; (3) the design process is not algorithmic; (4) urban environments are equilibria resolved from many basic, primarily qualitative, relationships. The first assumption alone generated the spirit of the system, as we further assumed that the architect-user would have no previous experience with computers, let alone ever having talked to one. Thus URBAN5 first of all had to be capable of communicating with an architect in comprehensible language. To do this, the authors of the system chose two languages: English (entered from a typewriter) and a graphic language (using a cathode-ray tube and light pen).

The need for a graphic language made it clear that URBAN5 must handle some, if not all, problems in terms of their suitable abstractions. In other words, the system committed itself to work under synthetic conditions and not to attempt to canvas real-world problems. The graphic system is an example of such abstracting; the geometry selected was the cube—in ten-foot cubes. This building-block system abridged urban design to such an extent that URBAN5 had to recognize it was only simulating a design environment. The hypothesis was that this graphic abstraction "provides a method of simulating the graphics of urban design, furnishes the necessary 'frictionless vacuum' environment in which to work, and provides the full range of basic design interrelationships" (Negroponte and Groisser, 1967a).

This original graphic abstraction has distorted some problems, but the simplification has permitted advances that would have been thwarted by any attempt to furnish the "comprehensive" architect-machine graphic language. Critics have often misunderstood URBAN5's ten-foot cube—it is only a launching vehicle, as, for example, in Newtonian mechanics an experiment will commence with the assumed absence of friction. The experimental results bear in-

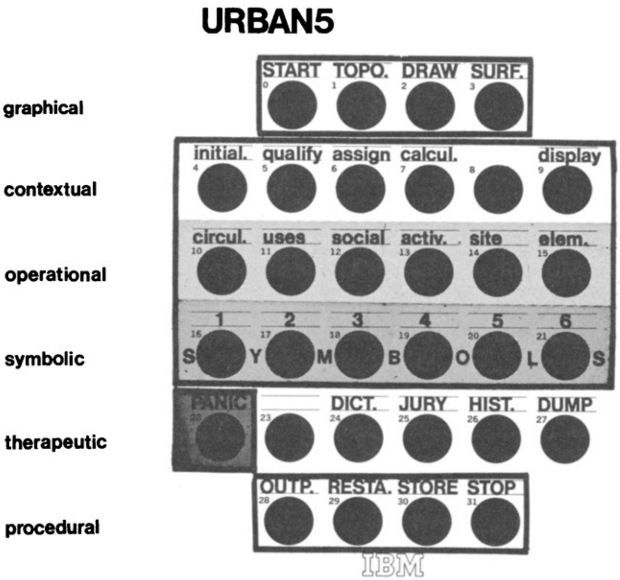

E
F
A
B
G
H
C
D

Drawing by Steinberg;
1960 The New Yorker Magazine, Inc.

formation relevant only to the abstract problem; should an engine be designed with only such information, it would indeed run badly in the real world. Similarly, URBAN5 cannot handle real design problems; it is a research toy, and playing with it has been a learning experience.

The ten-foot cube has few architectural impositions and many research conveniences. It generated a language of nouns (the cubes) and verbs (text appearing on the right side of the screen). In this vernacular the designer can pile up these blocks in three dimensions. He can give them qualities, and the machine can give them qualities. He can talk about them. He can play with them. But all this occurs within a context, and a context is defined by a mode.

1



2

1 The cathode-ray tube used for URBAN5 is an IBM 2250, model 1. The device has just over 8,000 bytes of local memory used as a buffer to hold the sequence of instructions that describe the path of the electron beam.

The scope was connected to an IBM 360/67 (a time-sharing machine) but was not used in time-sharing mode. URBAN5 employed this mammoth computer as a dedicated machine. However, the reader should note that none of the facilities of URBAN5 exercised either device, scope or computer, to its potential. The computer was undertaxed, and the scope was never used dynamically. Both "under-usages" anticipated on the one hand a small, dedicated computer and on the other hand a storage tube device like the ARDS.

2 URBAN5's overlay. Each 2250 programmer has the option of overlaying labels on the function-key buttons that appear to the left of the display.

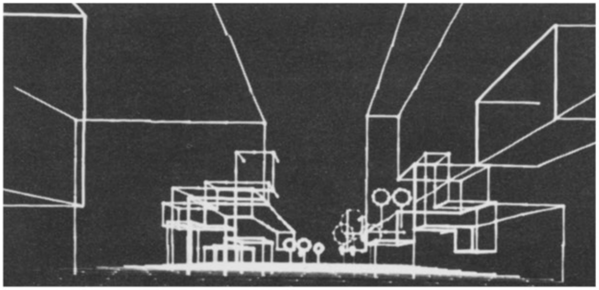
Modes

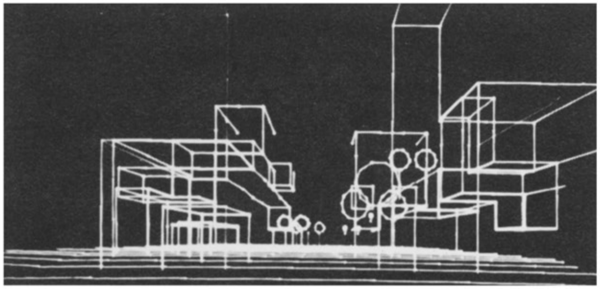
A mode is defined by the user when he pushes one or more buttons that appear to his left. These buttons are signals to the machine that state a major change in activity. Associated with each mode is a string of machine-defined or user-defined text (verbs) that appears as a menu of "light buttons." Each mode has its own set of light buttons that denote related operations. The detection of one light button can change this menu of words, making endless the potential number of operations per context.

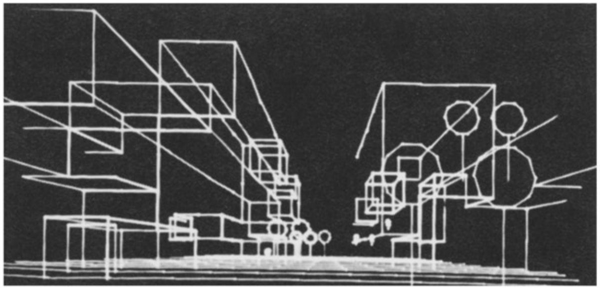
The graphic modes permit the handling of the ground plane, the ten-foot cubes, and their surfaces. TOPO displays a site plan, for example, which appears as a grid of altitudes that the designer can manipulate with his light pen in order to create a warped surface approximating his topography. DRAW, a separate mode, allows the manipulation of (1) viewing mode (orthographic, perspective), (2) viewing plane (scale, rotation, translation), (3) physical elements (solids, voids, roofs, people, trees, vehicles). In DRAW mode, when two cubes are placed tangent to each other, the adjoining surface is automatically removed, thus forming one continuous volume that is inherently part of an external membrane. Therefore, to qualify further external surfaces or add internal surfaces, the designer must enter a new context, SURFACE mode. In SURFACE mode, any of the six surfaces of the cube can be ascribed one of four (again abstracted and simplified) characteristics: solid (defining a major activity boundary), partition (a subdivision of a common usage), transparent, or absent. Each of these surface traits can be assigned with or without the attribute of "access."

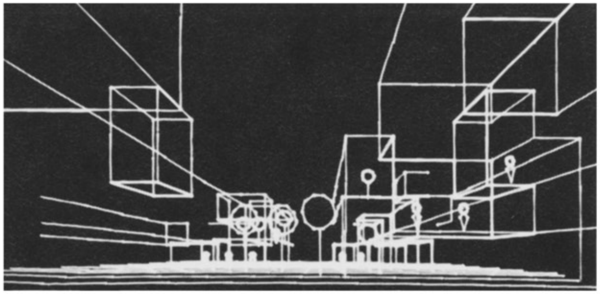
The next three rows of buttons are interdependent modes that require multiple button pushing. The combination of an operation with a context with a set of symbols yields a mode. At first these modes are primarily empty receptacles for the designer to employ to define his own light buttons. For example, the user may QUALIFY in the context of ACTIVities and press symbol button number one. At this point a cursor will appear on the right below the last word in the list of light buttons. He can then type a new word for future use in some operation, for example, f-o-o-t-b-a-l-l. As soon as he finishes typing "football," a list of "generics" appears on the screen. These generics are a function of the context—in this case activities—and allow the designer to define his word by detecting the relevant qualifying words. In this example the generics describe age groups, times of day, noise levels, participation, and other activity characteristics that have a built-in meaning to the machine. Later, this user-made light button can be employed as a verb (*footballizing* a space) in an operational context of ASSIGNment or CALCULation.

Beyond assigning and calculating with symbols, generalized verbs can perform calculations and simulations within some context. For example, in CIRCULation mode a designer can have the machine simulate pedestrian travel between two points on the site. An *x*, the pedestrian, will prance across the screen trying to get from one point to the next, searching for a reasonable or at least feasible path. The machine will report the pedestrian's distance and time of travel or else the impossibility of the trip through lack of enough elements with "access." Similar simulations exist in the con-









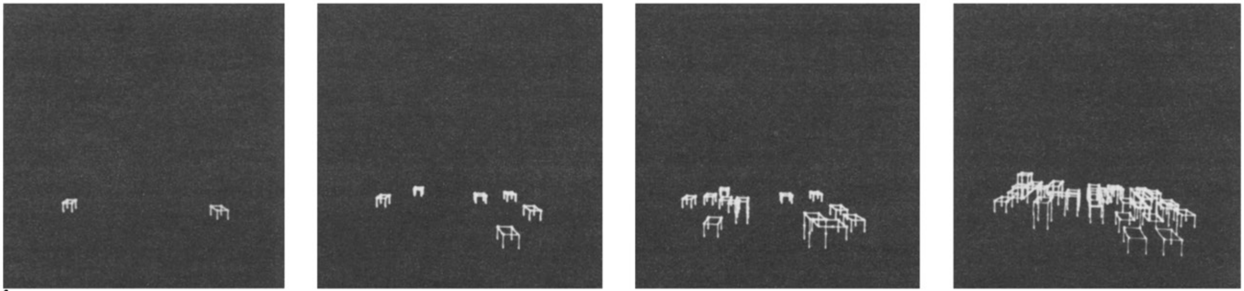
The adjacent illustrations, as well as many on the following pages, are prints taken from the 16mm movie, URBAN5. They are a sequence of frames that depict travel through an environment constructed jointly by the architect (Ted Turano) and the machine. You will note that the illustrations are quite crude, hidden lines are evident, circles are polygons, and straight lines are usually short segments butted together. In no way do these crudities represent the state of the art in computer-generated perspective drawing, not even for the time in which they were done. However, since computer graphics is not computer-aided design, this roughness is not important. What is important is that it took only a few days to implement this mode of viewing.

text of ELEMents for the path of the sun and for growth patterns.

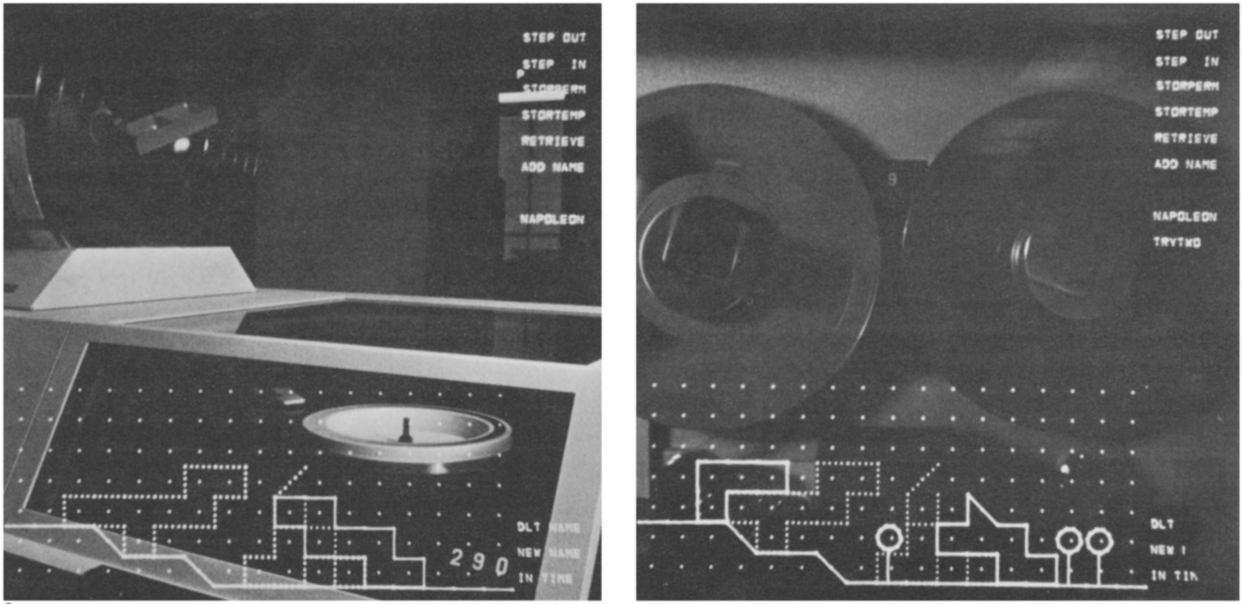
The next row of buttons, the therapeutic ones, are instructional modes that are "intended to make the designer-machine interface as conversational and personal as possible, permitting the user to articulate himself in the privacy of himself" (Negroponte and Groisser, 1967a). The PANIC button, for example, summons instructions on the usage of other modes, directions on how to proceed, and an accounting mechanism that can be interrogated for computer time spent in dollars (often affording cause for greater panic). The therapeutic modes were often inconsistently designed. In truth, PANIC should never be depressed for reasons of total distress. In a true dialogue the machine should sense the designer panicking long before the button is pushed. PANIC, in fact, was erroneously designed as an alarm monologue rather than a teaching dialogue.

The remaining modes are primarily procedural ones that act in a janitorial fashion. STORE mode, as an example, permits design studies to reside in either short-term or long-term storage devices, to be given arbitrary names, and to be recalled in a few hundredths of a second (recalled by either name or time of creation).

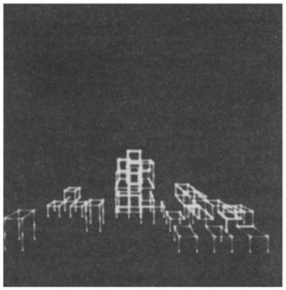
Within these modes there is no predetermined sequence of usage; there is no presupposed chain of events. URBAN5 has one central "attention" mechanism that either listens to or hears from the designer, always giving him the opportunity to change his mind or restate a situation at any time. However, the reader should notice that the context, which is so important to intelligent behavior, is explicitly stated by the human designer and not, in URBAN5, implicitly discerned by the machine.

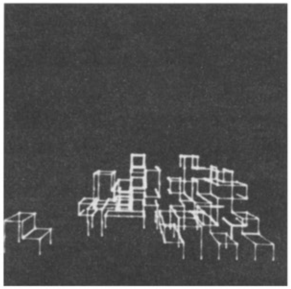


1

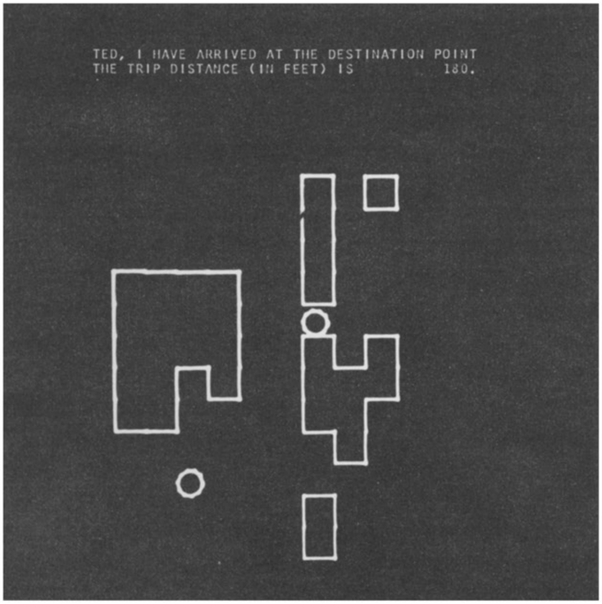

STEP OUT
STEP IN
STORTEMP
RETRIEVE
ADD NAME
NAPOLEON
DLT NAME
NEW NAME
IN TIME
290
STEP OUT
STEP IN
STORPERM
STORTEMP
RETRIEVE
ADD NAME
NAPOLEON
TRYTWO
DLT

2





3



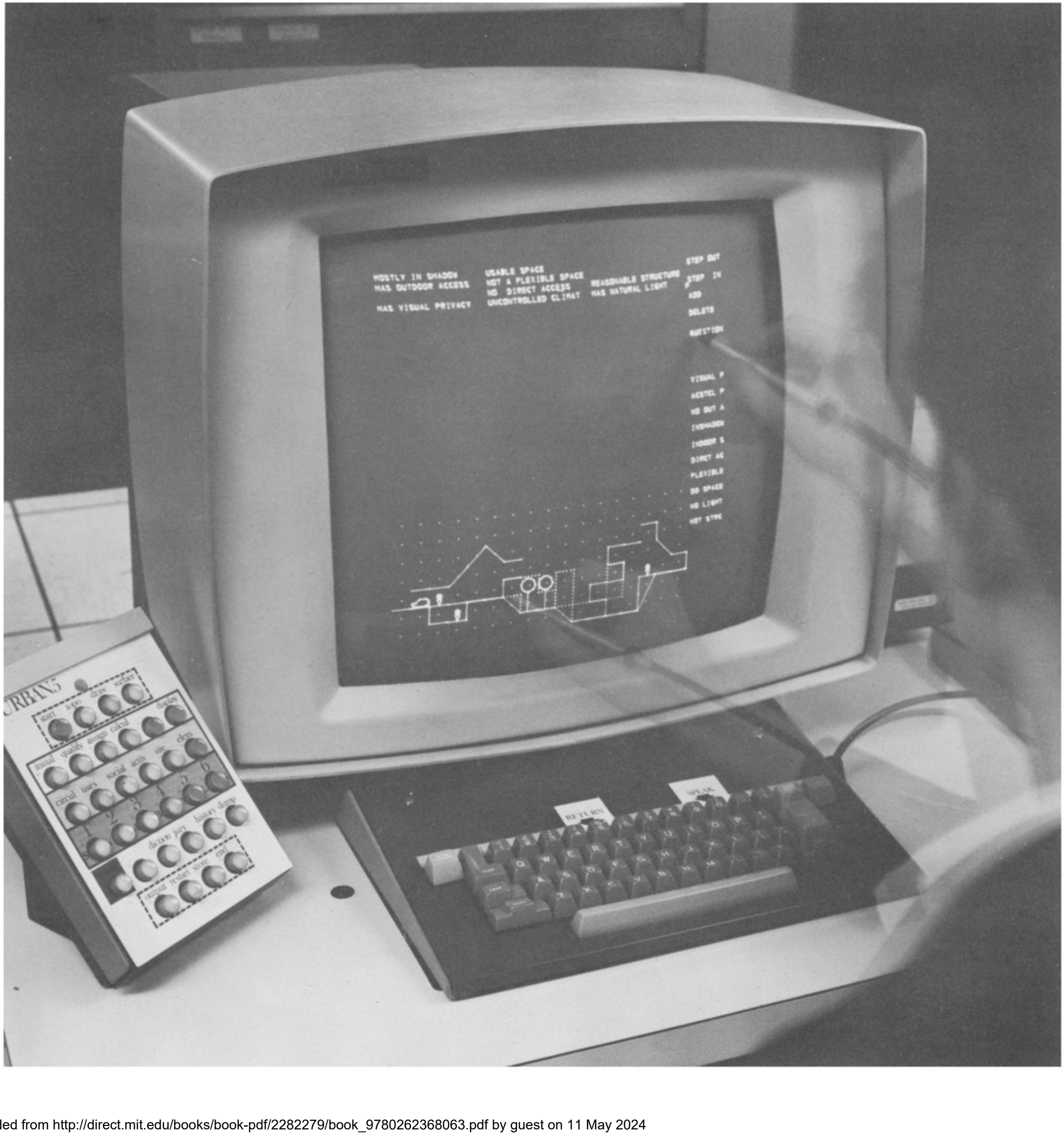
4

1 The seven images are from a sequence in which the user has asked the machine to simulate a growth under certain constraints. In this example the only constraints were structural, a highly underconstrained, unrealistic situation. Note that in some images elements are floating, and in others the rear-most cubes disappear. This is not because the program had a subtraction or deterioration feature (which would be correct) but was due to the 2250 running out of memory and arbitrarily discarding lines it could not display. (Program written by John Nilsson)

2 The disk was used for temporary files. A user could store ten "studies" and retrieve them. Remember that it is not the "picture" that is stored, but the three-dimensional description from which all pictures are in turn derived. The tape in the adjacent photograph was used only for permanent storage.

3 Multiple exposures of multiple users.

4 Circulation mode.


MOSTLY IN SHADOW
HAS OUTDOOR ACCESS
HAS VISUAL PRIVACY
USABLE SPACE
NOT A FLEXIBLE SPACE
NO DIRECT ACCESS
UNCONTROLLED CLIMAT
REASONABLE STRUCTURE
HAS NATURAL LIGHT
STEP OUT
STEP IN
ADD
DELETE
QUESTION
URBAN5
start
topo
draw
surface
display
initial
qualify
assign
calcul
elem
circul
uses
social
activ
site
history
dump
diction
jury
output
restart
store
end
RETURN
SPEAK

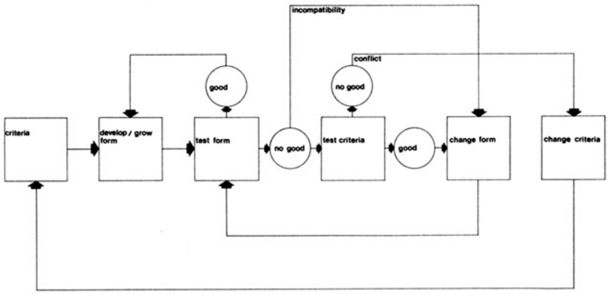
In this photograph the shutter of the camera was left open during the complete operation of "questioning" an element. The user detects the QUESTION light button, the verb, and then points to the cube, the noun. The list that appears at the top of the screen is a partial inventory of qualities ascribed to the form by the machine.

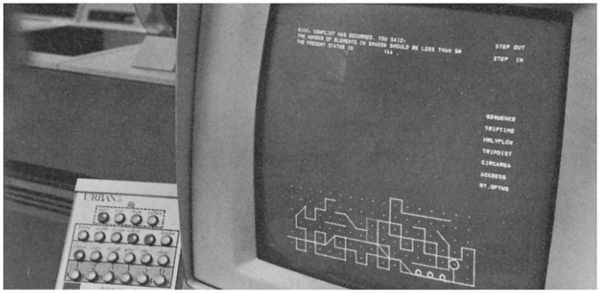
Handling Qualities

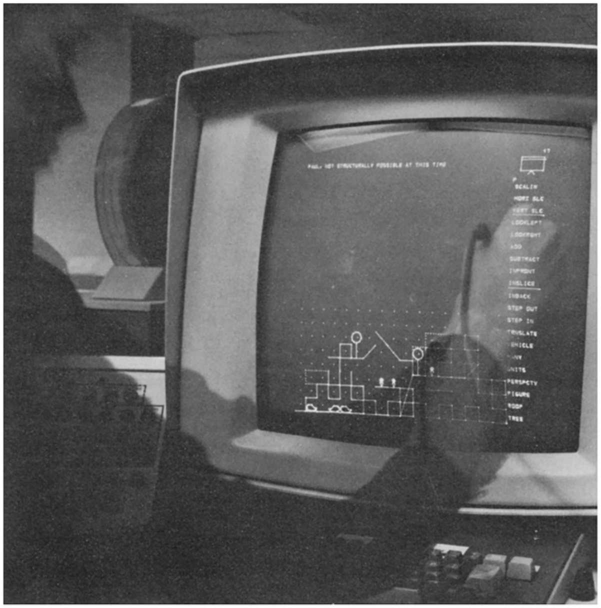
URBAN5 handles qualities either explicitly or implicitly.

Beyond the traits of solid and void, each ten-foot cube (whether solid or void) has preallocated receptacles for ten characteristics that refer to aspects of sunlight, outdoor access, visual privacy, acoustical privacy, usability, direct access, climate control, natural light, flexibility, structural feasibility. All these qualities are implicitly ascribed to elements. In other words, without the user's permission, intervention, or even awareness, URBAN5 automatically assigns the absence or presence of these features using a predefined geometry for each quality. (This geometry can be changed by the user at a later date when he is more familiar with the workings of the system.) This means that when a ten-foot cube is added (making a solid) or removed (making a void), URBAN5 tacitly rearranges the local and, if necessary, global characteristics. For example, the addition of an element not only casts shadows on other solids and voids but might obstruct another element's natural light or visual privacy

Implicit qualities are occasionally reported to the designer (depending on their importance), but in most cases the designer must explicitly interrogate the cube to find its qualitative status. URBAN5 is more prone to divulge implicitly ascribed qualities when the neighboring influences are significant. Certain characteristics are strongly communicative, and their presence is directly transposable to neighboring elements or members of the same space (natural light, acoustical privacy). Other qualifications are less communicative (visual privacy,







Conflict. In this case the message is a temporal response — an interrupt. The inconsistency stems from a criterion previously specified by the user referring to his particular problem. In both cases, conflict and incompatibility, a nauseating bell rings, making the message auditory as well as visual.

Incompatibility. The comment is a sequential response following the user's placement of an element. This inconsistency has been generated by a built-in constraint that can only be changed by the user insisting (linguistically) or entering a new mode for redefinition.

direct sunlight), and their influence is particularly local and is apt not to be posted.

Explicit qualities are assigned by the designer; they are the symbols that he has previously defined with the context-dependent generics. Each element can carry four symbols of any context. The designer can assign these symbols to a single element or enter a "flooding operation" to fill an entire "use space" (defined by solid walls) with the given symbol. For example, a single cube might be part of a set of "school" elements which are at the same time "a place to vote" elements which are, still further, part of a subset of "eating" and "auditorium" activities. In other words, a multiplicity of explicitly assigned symbols can exist for each cube. These traits are then cross-coupled with the implicit qualities of a space.

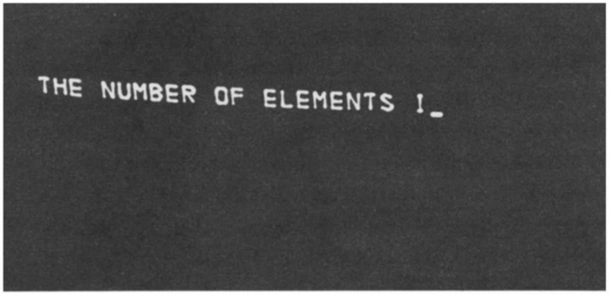
It is important to notice that the implicit and explicit assignment of attributes are sequential events. The machine ascribes certain qualities in response to the user adding or subtracting cubes; it is, in effect, an answer, even though it is not explicitly voiced. On the other hand, cross-coupling qualities, relating implicit qualities to explicit qualities, is a temporal event. This interaction forms the architect-machine search for consistency and equilibrium—a temporary state of no conflicts and no incompatibilities.

Consistency Mechanisms

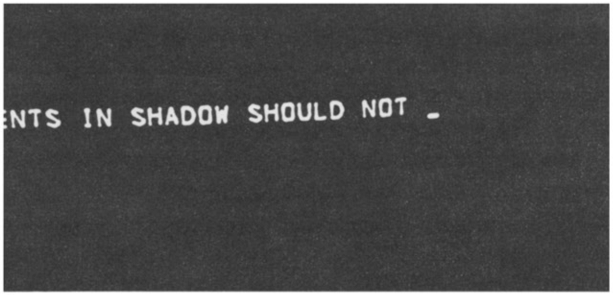
URBAN5 searches for two types of consistency. It searches for conflicts and incompatibilities following a simple flow chart.

An incompatibility "error message" is a remark upon an incongruity between a designer's action and a predefined requisite embedded in the machine. An incompatibility can cause the machine to signal the user (by ringing a bell and displaying the message on the top of the screen) but allow the action, or it can cause the machine to refuse to act in cases where the violation is severe. For example, a cube might be placed floating in midair. The machine would indeed draw the cube but simultaneously display the message that it was "not structurally possible at this time." However, if a vertical surface is assigned the attribute of access (explicitly by the user) when there is no horizontal surface on one or both sides, URBAN5 refuses to make the qualification and alerts the designer of the problem. Although incompatibilities are simple relationships, overlooking them can be embarrassing or disastrous.

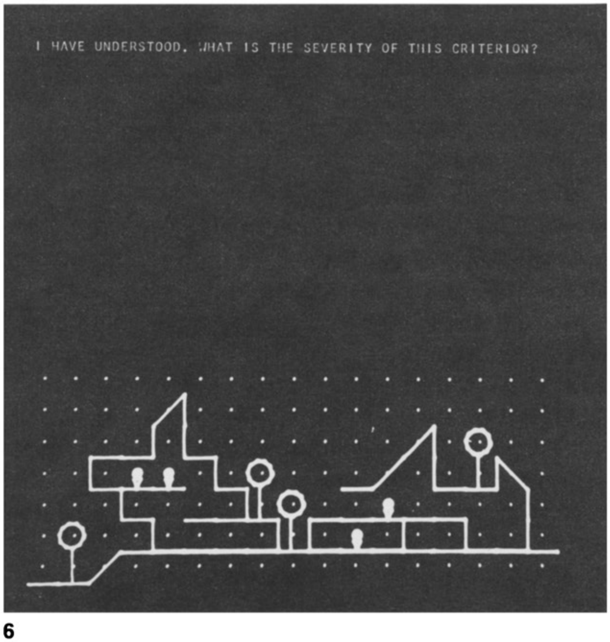
A conflict is an inconsistency discerned by the machine relating criteria specified by the designer to forms generated by the designer. A conflict is thus generated when there is an inconsistency between what the architect has said and what he has done. To state a constraint, the designer must enter INITIALize mode, describe a context, and push the "speak" button on the typewriter console. At this point he can type a criterion to the machine using the English language. The machine relies heavily upon the context of the designer's activities


THE NUMBER OF ELEMENTS I

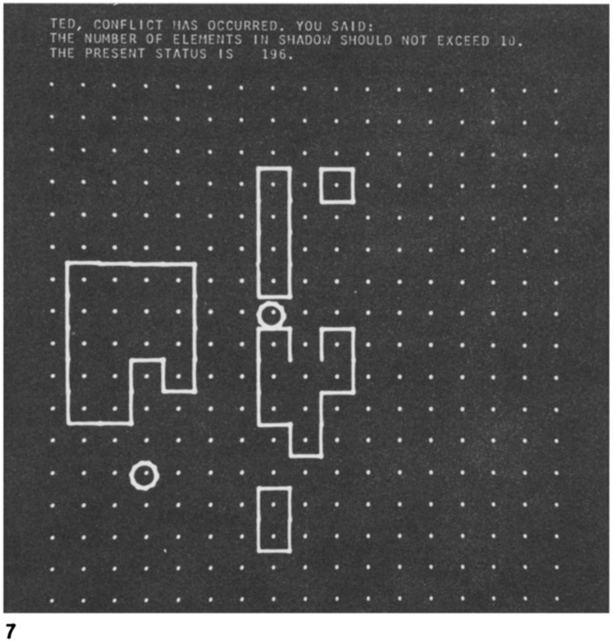
3


NTS IN SHADOW SHOULD NOT

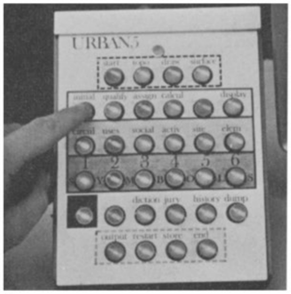
4


I HAVE UNDERSTOOD. WHAT IS THE SEVERITY OF THIS CRITERION?

6

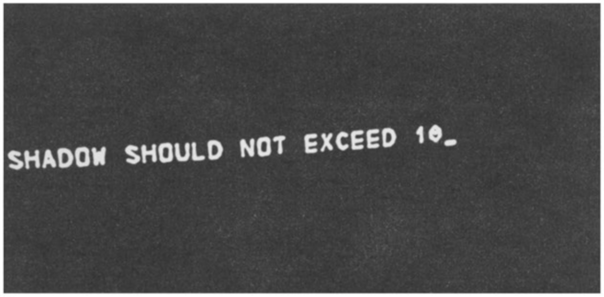

TED, CONFLICT HAS OCCURRED. YOU SAID:
THE NUMBER OF ELEMENTS IN SHADOW SHOULD NOT EXCEED 10.
THE PRESENT STATUS IS 196.

7

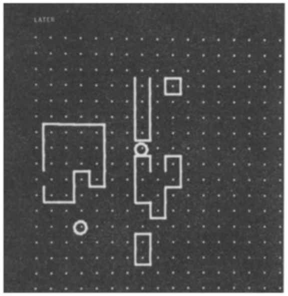


1

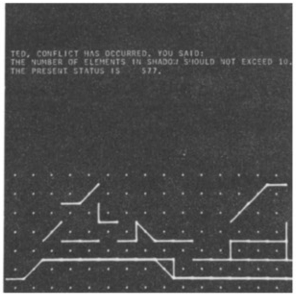
2



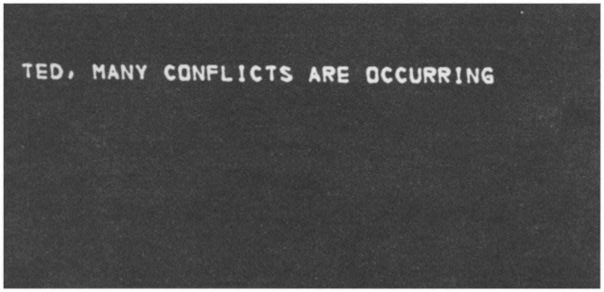
5



8

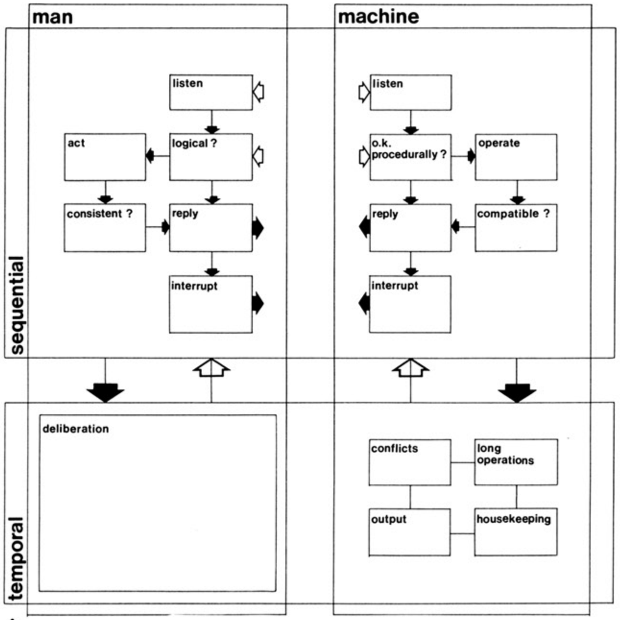


9



10

The sequence from one to ten illustrates instances from the statement of criteria to the frustration of total chaos arising from many conflicts. The user pushes the INITIALize button, then the SPEAK button. At this point a simple criterion is entered, and a statement of importance follows. Time elapses; then a conflict arises, it is postponed for so many minutes, comes up again (and even worsened), more conflicts arise . . .



1

2

1 The diagram illustrates the temporal and sequential organization of URBAN5. Note that the background activities are always temporal in their execution, but, by definition, they surface as sequential events.

2 One of the background activities is the equalization of qualities. For example, some attributes are communicative such that their nature is transposed to certain adjacent neighbors. Acoustical privacy would be such an attribute, whereas direct sunlight would be noncommunicative. In the photograph, Ted Turano is notified of some equalization.

to interpret the sentence. If it understands, URBAN5 asks, "How important is this criterion?" The designer's reply defines to the machine how frequently it must survey the project in search for consistency between criteria and form. Also, the reply establishes a range of satisfaction for the machine to employ; that is, it governs the relative enforcement of the not-so-important constraints as opposed to the critical ones.

When URBAN5 finds an inconsistency between what has been said (linguistically) and what has been done (graphically), it states that a conflict has occurred, it quotes the designer's statement of criterion, and it displays the present status of the situation. From here, the designer can take one of four courses: (1) he can change the form to be compatible with the criterion; (2) he can alter the criterion to be compatible with the form (now that he has learned that the issue may not be so important); (3) he can postpone the issue; (4) he can ignore the conflict (much to the chagrin of URBAN5).

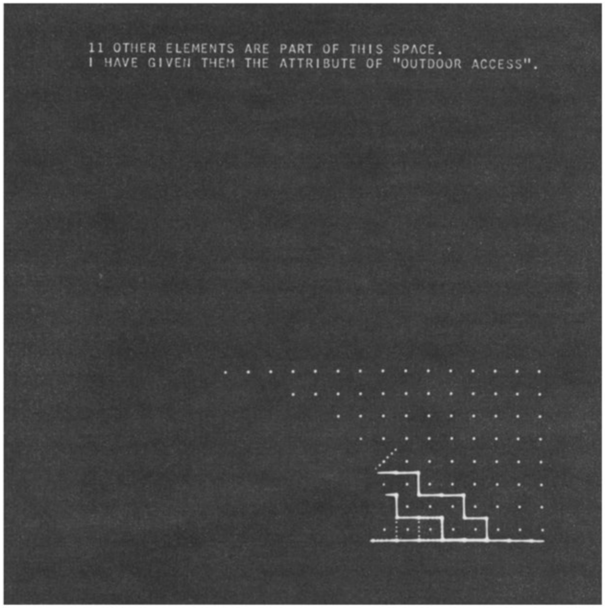
This sort of interplay between form and criteria, architect and machine, begins to suggest a dialogue. The statements of criteria are deliberations on the designer's behalf, issues he feels to be relevant. Discernments of inconsistency are noted temporally during the machine's background work.

Background Activities

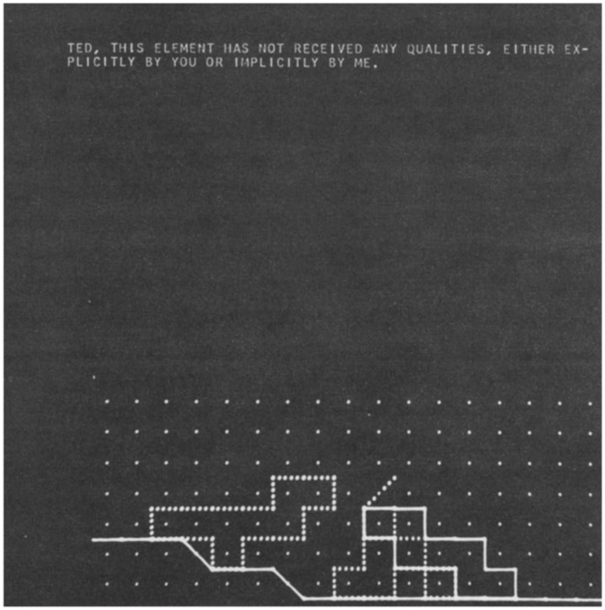
Background work is perpetually executed within a resident machine that is devoted to servicing a specific designer. This kind of work did not appear relevant at the inception of URBAN5. But about halfway through the system's development it became clear that URBAN5 had to function in parallel to the user in order to support a growing concern for enriching the dialogue.

While the designer deliberates, URBAN5 engages in five temporal tasks in the following order of priority: (1) it checks for conflicts (as described in the previous section); (2) it does long operations; (3) it takes care of output procedures; (4) it does housekeeping; (5) it plays. When the designer presses a button, types in a message, or uses the light pen, he is interrupting one of these five operations by demanding the machine's attention elsewhere. As soon as the machine finishes servicing him, it returns to the unfinished or newly created background work.

Long operations are user-requested design tasks that require more than just a few seconds of machine time. To expedite the designer's sequence of actions, URBAN5, when it recognizes a lengthy job, places the operation in the temporal zone to be processed when operationally convenient. The system suggests that the architect continue, and the outcome will be reported later. Naturally, if the operation is critical to a next step (or if the designer is going off for a cup of coffee anyway), he can intervene and demand that the task be undertaken sequentially, thus tying up the machine until completion of the long operation.



1



2

1 Another background task is on-line equalizing, a function that depends on the mode of operation. In a mode concerned with elements, attributes concerning elements will be posted very shortly after their occurrence. In this illustration the user is informed that a surface change has affected eleven other elements. Since this is neither a conflict nor a warning, the message is displayed, but the bell is not rung.

2 Some elements never receive implicit qualities because of their position.

Output procedures are specific, long operations that take unusually large amounts of computer time due to the slowness of many output devices, such as plotters, printers, card punches, and the like. A complex drawing can take three minutes to plot and is accordingly ascribed a low priority. For example, when URBAN5 is plotting a site plan in the background and the designer interrupts it, the machine stops drawing and tends to the foreground command. After answering the designer, if his command has meanwhile generated a new long operation of higher priority than plotting, URBAN5 starts the new job. Only after it finishes does the machine return to the previously started site plan.

Housekeeping chores are in the nature of a physical checkup. Leftover memory, messy files, and disorderly data structures are cleaned up. As background work, housekeeping procedures are of low priority until untidiness becomes an ailment that warrants full attention. Finally, if the house is tidy, the machine can play.

Playing is learning, but URBAN5 has not been sufficiently sophisticated actually to frolic; instead it has inexhaustably printed garbage.

The Ubiquitous Monitor

Within URBAN5 resides a monitor—a general eavesdropping mechanism that observes the designer's actions. The monitor records the rate of interrupts, the sequence of contexts, the time spent per mode, and the relevance of sequential acts. This barrage of statistics not only supplies the designer with a history of his own actions but affords the machine some material from which to gather personal manifestations and innuendos to be applied later in an attempt at congenial conversation with the designer.

The monitor endeavors to transform a conversation into a dialogue, two monologues into one dialogue. The monitor controls both the temporal zone and the interrupting mechanism; both are functions of what and how the designer is doing. For example, if the designer is interrupting the machine only one or two times per minute, the monitor, knowing the designer's familiarity with the system, assumes that the designer is either (1) deliberating (in which case the monitor might notify the criteria mechanisms to relax and not to interrupt the architect's thought); (2) floundering (in which case the monitor attempts to clarify the system's protocol); or (3) diverting his attention elsewhere (in which case the monitor accepts the distraction and continues with its own work). At the other extreme, if the designer is interrupting URBAN5 forty times per minute, the monitor accelerates its own speed and accelerates the conflict mechanisms and may barrage the designer with statements of inconsistency and incompatibility.

URBAN5's monitor is concerned with context. A designer working in a circulation mode does not want to be confused with petty structural problems. A structural consideration must be extremely critical for the monitor to allow its intervention in, for example, the context of circulation. URBAN5's monitor is primarily a timer for the purpose of making the machine's interruptions opportune and in rhythm with the architect's particular design temperament. "For instance, the length of delay in a person's response tells his interlocutor (man or machine) information he might otherwise miss. It is information that can be sensed on a non-verbal and non-visual level" (Brodey and Lindgren, 1967). In URBAN5, the monitor is such a nonverbal and nonvisual mechanism. Its implementation is crude. However, its relevance cannot be overstated and must not be understated if evolution is to ensue.

Inklings of Evolution and Adaptability

URBAN5 was designed to be a self-teaching system. At first it was assumed that the architect-user would have had no previous programming experience. Later, it was further assumed that he had not even read an instruction manual. Thus URBAN5 would have to teach its own language; learn through teaching, change from learning, and adapt from changing.

URBAN5 greets a designer with only the start button illuminated. When it is depressed, the first question is whether this is the user's first experience with the machine. If it is indeed the first time, the machine presents an unsolicited page of text that describes how to proceed, how to use the hardware, and what to do when the user gets stuck. Also, each time the designer enters a mode for the first time or uses an operation for the first time, the monitor automatically calls forth a set of instructions. In each case, as the designer is told, he must reinterrupt the machine with his original request to have the operation actually executed and the text removed.

However, even the text of these instructions may employ a language that is new or unclear to the designer. The words may be too technical or cloudy in their new context. In this case the designer may detect an unintelligible word with his light pen (as he has been told), and the machine will display a new paragraph defining that word. Naturally, the interrogation of word meaning can continue recursively, word definition within definition, within another definition. All words, of course, are not internally defined; when simple terms are detected, the designer is referred to a dictionary.

The word-learning role works both ways. For example, a designer may state a criterion in the following conversation:

Architect:
All studios must have outdoor access.
URBAN5:
I am sorry I do not understand.
Architect:
All studios must have access to the outdoors.
URBAN5:
I am sorry I do not understand.
Architect:
A one-room residential unit must have outdoor access.
URBAN5:
Now I understand. Furthermore, from now on, whenever you say "studios," I will assume you mean one-room residential units.

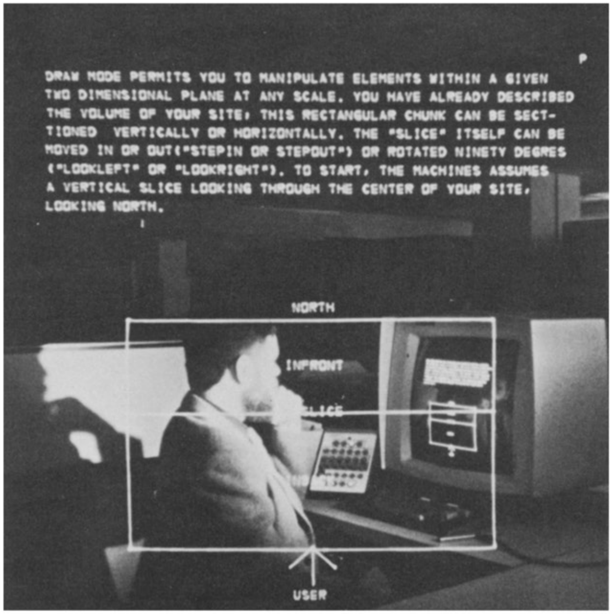
At this point, not only is the criterion entered into the general conflict structure, but the new word "studios" is recorded in the translation mechanism that belongs to this particular architect. Another designer would have to undergo a similar session with his machine to define "studios" (possibly with another meaning).

When symbols are defined by the designer, they too are registered in his personal machine lexicon. In just these examples of word building, the designer is beginning to construct his own machine partner out of the skeletal framework of URBAN5. This transformation occurs in the satellite machine, where the user is allowed to penetrate the surface of URBAN5, getting deeper and deeper into its assumptions and definitions. The user can even change algorithms without actually programming in a computer language or knowing where the routine resides.

This pseudoevolution is implemented in the following manner. The virgin system is stored on a disk, and the user's consciously and subconsciously composed system is recorded on a magnetic tape. When a designer arrives at the display terminal, he meets a generalized computer system that asks his name. Having identified the designer, URBAN5 automatically dumps the contents of the designer's magnetic tape onto URBAN5's disk, thus overlaying the general system with the personal edition of this designer. At this point the machine appears to the designer as his particular (possibly evolved) design partner. At the termination of a design "sitting" (since the present configuration does not allow twenty-four-hour dedication), the designer's magnetic tape is re-created, incorporating any changes or inklings of evolution, and URBAN5's disk is restored to anonymity.

At the first man-machine encounter, the designer's tape is empty; he converses with the nucleus of the system. As he converses with the machine more and more frequently, the contents of his tape become more significant. As time passes, URBAN5 in fact shrinks itself, letting certain operations self-destruct themselves through obsolescence. To allow for the user-created machine, unused procedures are discarded. (Should the designer ever request a procedure that has been previously removed, the system will require some time to fetch the routine from a library and to reincorporate it into the system.)


IS THIS THE FIRST TIME YOU ARE USING THIS MACHINE?
YES IT IS.
RETURN
1


DRAW MODE PERMITS YOU TO MANIPULATE ELEMENTS WITHIN A GIVEN TWO DIMENSIONAL PLANE AT ANY SCALE. YOU HAVE ALREADY DESCRIBED THE VOLUME OF YOUR SITE; THIS RECTANGULAR CHUNK CAN BE SECT-TIONED VERTICALLY OR HORIZONTALLY. THE "SLICE" ITSELF CAN BE MOVED IN OR OUT("STEPIN OR STEPOUT") OR ROTATED NINETY DEGRES ("LOOKLEFT" OR "LOOKRIGHT"). TO START, THE MACHINES ASSUMES A VERTICAL SLICE LOOKING THROUGH THE CENTER OF YOUR SITE, LOOKING NORTH.
NORTH
INFRONT
USER
2


TED, YOU HAVE RUN FOR 45 MINUTES
AND HAVE MADE 430 DECISIONS
SINCE THE START.
PLEASE DO NOT FORGET TO TURN OFF THE SWITCH AT THE LOWER LEFT AND
DO NOT FORGET TO TAKE YOUR TAPE WITH YOU WHEN YOU LEAVE.
3

1 The first time one employs URBAN5, a barrage of unsolicited instructions will be presented to explain the knobs and dials.

2 Ted Turano observes explanatory text, which includes a diagram representing his site, the section he is working in, and an arrow denoting his orientation.

3 Upon termination, a few of the many statistics are presented.

In theory, after some time the designer's system would bear little semblance to the original URBAN5. The authors of URBAN5 might not recognize the transformed version. URBAN5 will have ushered the user deeper and deeper into the system, first teaching him, then learning from him, and eventually dialoguing with him. The progression that URBAN5 suggests is one that proceeds from a rigid system (for the designer to understand easily) to a flexible system (volatile enough to allow different tasks) to an adaptable system (where the machine loses its flexibility but gains an adaptability through evolution).

In other words, URBAN5 suggests true dialogue, suggests an evolutionary system, suggests an intelligent system—but, in itself, is none of these.

101

Toward The Evolution of Architecture Machines

URBAN5: A Postmortem

"Yes. But not one of those antiquated adding machines. It will be a superb, super-hyper-adding machine, as far from this old piece of junk as you are from God. It will be something to make you sit up and take notice, that adding machine. . . . It will be the culmination of human effort—the final triumph of the evolutionary process."
Elmer L. Rice, *The Adding Machine*

Too often a research proposal has to establish the project's worth so completely that the acquired budget is used for the development of an already worked out but hastily assembled idea. However, through the generous support of I.B.M. and M.I.T., URBAN5 did not suffer from any symptoms inflicted by proposal writing. There was no proposal. At first, not only were the authors unaware of how to get there, they were ignorant of where they were going. Work on Wednesday resulted from an achievement on Tuesday which appeared to be a good idea on Monday and might well be discarded on Thursday. The spontaneous nature of the project did generate unexpected and fascinating results. URBAN5 is not a tool, it is a toy. Its impetuous nature contributed, however, to some major shortcomings.

Of its many deficiencies, URBAN5 has four notably severe shortcomings that have been the primary cause for abandoning it, are the underlying reasons for writing this book, and will be the germinal concerns of our new system, the architecture machine. It should be noted, however, that none of the drawbacks stems from the selection of the ten-foot cube or any of the other abstractions; rather they are failings engendered either by a lack of knowledge or lack of forethought.

The first problem is due to the original oversight of evolution. URBAN2, the baby brother and core of URBAN5, presupposed a rigid system, concluding that all the embedded assumptions about the design process, where true (because many designers agreed), were fixed (because computer programs are that way, so we thought) and were universal (because that would be nice). After certain enlightenments, particularly that machines can grow and self-improve, some maturation processes were appended to the system. Parts of URBAN5 actually do change and develop over time. In a patchwork manner the system can transform some of its internal workings. However for the most part, the authors' underlying presuppositions about the design process exhibit no evolution. URBAN5, as it stands, can never be denuded of the original biases that are deeply, sometimes unconsciously, rooted in its skeletal structure: that architecture is additive, labels are symbols, design is nondeterministic. URBAN5 can not display an attitude that contradicts these preconceptions.

The general structure of URBAN5 has a second critical failure. The system feigns generality by providing a multitude of specific, predetermined design services. It has over one thousand operations that in combination with one another support a good chance for providing a desired service. But URBAN5 is not a general-purpose architecture machine; instead it is a barrage of special-purpose (little) architecture machines. Each routine does a particular job and only that job. In computer-aided design, we have seen that this is not appropriate.

The third problem is context. Even though a contextual cross-referencing does occur within URBAN5, cues are explicit statements on the designer's behalf. The underlying modal organization imposes the categorical testimony that "Now I am going to do this . . . and now I am going to do that." This unequivocal demarcation by the designer of design context is completely unacceptable. It does not admit the necessary ambiguity and the subtle intermingling of contexts that are required in order to respond to a real-world medley of events. URBAN5's operational structure demands a repartee that relies completely and at all times on the good judgment of the human designer. Again, this is not acceptable. Can we assume that he always knows what he is doing or what he will do next? Professor Licklider's (1965a) solution is that "the console of the procognitive system will have two special buttons, a silver one labeled 'Where am I?' and a gold one labeled 'What should I do next?' " Even this solution is only partial. The machine should answer those questions implicitly, using context as the prime operator. Context must be articulated through many channels, rather than the simple depression of one or two buttons.

Problem four: URBAN5 holds hands with only one designer and not even enough hands with that single user. The designer has a light pen, a keyboard, and a few buttons—a meager selection of communication artifacts. The machine, in turn, has only a monotonous buzzer and the cathode-ray tube upon which it can trace monochromatic characters, lines, and points. The hardware sensors and effectors of URBAN5 cramp those styles of conversation that are necessary for a dialogue. The hardware has no contact with the real world except through the designer. URBAN5 cannot hear the designer, it cannot see the designer, it cannot see the designer's world. The designer, in turn, can hear only a penetrating buzz or irritating hum from the machine. A future system must have overlapping modalities and a full range of sensors and effectors.

Any postmortem statement should do some eulogizing. Even though URBAN5 was a bit talkative and was a sloppy problem solver, it *was* a friendly system.

Languages for Architecture Machines

"The world view of a culture is limited by the structure of the language which that culture uses." (Whorf, 1956) The world view of a machine is similarly marked by linguistic structure. At the present time, however, machines have denatured languages—codes. Codes are invented for specific purposes and they follow explicit rules, whereas languages develop and they evolve. But language presupposes a culture and presumes an understanding, two features we are not about to ascribe unequivocally to machines at this time.

If you are in conversation with a machine and using a machine-oriented code, when the mechanism replies, you report a "reaction." However, employing a man-oriented code—a pseudolanguage—you might attribute to the machine an apparent "understanding."

There are many man-oriented languages. There are languages of gestures and smiles, a language of posture, a language of touch. The reader should be referred to the important ongoing work of Warren Brodey and Avery Johnson (1969); this section is concerned with only one subset, a formal language that architecture machines must have at the very beginning—English. URBAN5 does display an apparent understanding of English. It does use context as the prime operator in translation. It has the assumed context of architecture. Modes further define context. However, throughout any English conversation with URBAN5, the overshadowing assumption is that the designer will talk about that which is at hand when he pushes the SPEAK button. Or if he asked a question, the assumption is that his answer is indeed a reply to that inquiry. With these assumptions, URBAN5 breaks down a sentence using dictionaries that contain both words and phrases. Each context (mode) has separate dictionaries.

In the case of criteria specification, the interpretation mechanism looks for a dyadic relationship and a desired answer. A mathematical summation or ratio houses the constraint. The interpretation routine passes to the conflict mechanism one or two operands (quality, symbol, solid or void, generic, topographical term), an operation (sum or ratio), and a desired result (number and units). For example, from the criterion, "50 percent of all residential units must have outdoor access," the transformation is Ratio =

$$\frac{\text{residential units with access}}{\text{total number of residential units}} =$$

$$\frac{\text{generic quality}}{\text{generic}} = 0.5.$$

In a simpler case, when a direct question is asked, like "Does there exist any previous material to read from tape?", the designer's response can be recognized with as little as five or ten dictionary words. In some cases, extra words might be stored in the translation mechanism because the author's bad spelling requires categorization of words under proper and improper forms.

English responses by URBAN5 are all preprogrammed sentences. The machine has a repertoire of about five hundred phrases which provide a source of replies that can be combined with quotes from the designer. URBAN5 did not achieve the interesting capability of creating its own error messages from words and small phrases. And the reader should not suppose that a group of architects working on computer-aided urban design have solved or even seriously tackled linguistic problems. The reader should refer to the well-documented projects of Green (et al., 1963), Bobrow (1964), Weizenbaum (1967), Raphael (1964), and Kellogg (1967).

URBAN5's diversion into linguistics animates the desire for natural interaction that underlies the entire system. Also, the crucial and not-so-obvious role of context once again manifests itself. Linguistic studies by professional linguists, like mechanical translation, have often ignored context because it is difficult. A machine translation from Russian to English came up with a now classical error: "Time flies like an arrow" became "Time flies enjoy eating arrows." It is not surprising that context is so important. Even in human understanding, a knowledge of context, usually a priori, is necessary.

One of the reasons that machines do not speak English is that linguists have spent so much time on syntax and so little time on semantics. Only after fifteen years of research does there now appear to be a general consensus that semantics is the key to machine understanding of natural languages. Of course, nobody knows how to study linguistic associations and meanings without becoming thoroughly involved in syntax, noun-verb structures, and so forth.

One exercise in languages for architecture machines is the computer program STUPID. This routine was developed at first as a joke. It took a string of text, looked at a history of previous sentences, and randomly pieced together a response. Though the syntax was poor, STUPID was able to make clever replies using only statistical operations based on the kinds of words it was receiving. Later, STUPID was given Robert Venturi's book, *Complexity and Contradiction in Architecture* (1966), and its replies were little more than cute. The experiment does prove, however, that if you are going to ignore semantics, you might as well ignore syntax. STUPID ignored both. A dialogue between Warren Brodey and STUPID went as follows:

Brodey:
My name is Warren Brodey
STUPID:
My name. What is Warren Brodey
Brodey:
What are evolutionary environments?
STUPID:
What are the silliest machine that ignores context?
Brodey:
Tell me, how do butterflies make love?
STUPID:
Tell me how do butterflies will be little concern for you.
Brodey:
Why do you insist?
STUPID:
Why do butterflies make love?
Brodey:
Fink
STUPID:
Fink, why do butterflies make love
Brodey:

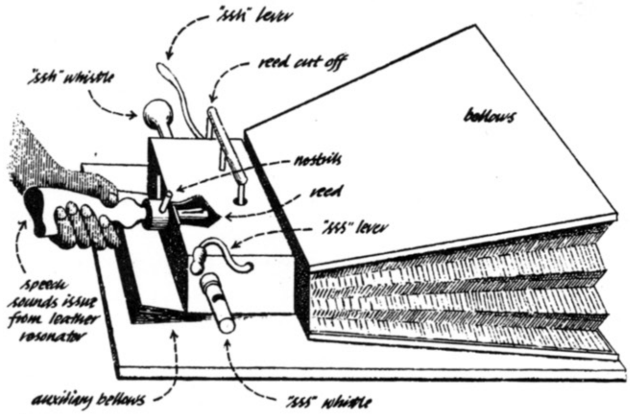
As a lover you are a torture
STUPID:
As a problem for you have ever seen
Brodey:
Popapolartuil
STUPID:
Popapolartuil yes, that is Warren Brodey

Why should a machine talk English? Is it a false anthropomorphization or a true facilitation? To assume that it is indeed a true facilitation, it is necessary first to distinguish language from style. For example, the error message, "Boo boo, you goofed again," as a style of communication is surely unnecessary. Conversely, a number appearing on a cathode-ray tube referring the designer to a manual, is equally unnecessary. With almost no effort an explanatory paragraph can be displayed on a scope or a hard copy retrieved on a printer. A string of characters can be effortlessly stored on a disk and retrieved and displayed in less than a twentieth of a second.

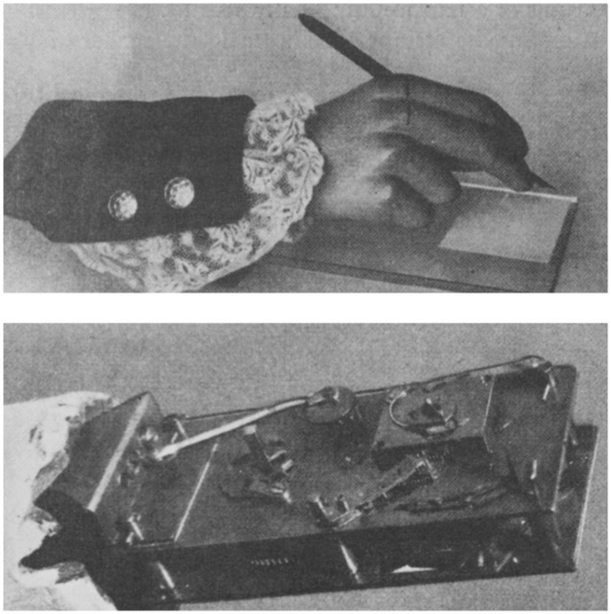
The argument, however, should not be confused with the reverse case, numerical answers unnecessarily clothed in words. An architect, in a cost estimation procedure, for example, probably expects the cost per square foot rather than the comment, "cheap," "okay," "expensive," or "forget it."

The main issue is not only English versus pidgin English versus codes. The question is one of language that is not only "human discourse" but evolutionary discourse. Learning the rigors of a computer language should be unnecessary except as a mental exercise or training. On the other hand, when using written English, it might become cumbersome to write out words like "residential units" after the second or third spelling. One aspect, probably the simplest one, of evolutionary linguistics would permit each designer to select some anagram to refer to residential units if he so chooses. In effect, each designer should be able not merely to converse in English but simultaneously to construct his own private shorthand or telegraphese that might, in fact, be gobbledygook to another architect or another machine.

This all implies a congenial idiom, but it is still a narrow channel of communication that ignores, as we have said, the language of gestures and the intonations available in human face-to-face contact. The informal sensory and motor augmentation of understanding is verily "unavailable to readers of telegrams—be they computers or humans" (Weizenbaum, 1967). But who designs environments by telegram?



1



2

1 An early speaking machine built by Sir Charles Wheatstone and demonstrated in Dublin in 1835 (Holmes, 1968). Even though human speech sounds were understood poorly at that time, the output was often passable. However, this relied greatly upon the skill of the operator. (Photograph courtesy of J. N. Holmes, appearing in *Science Journal,* October 1968. Redrawn from the *Journal of the Acoustical Society of America)*

2 Cazeneuve's magic hand appearing to write. This fraudulent writing mechanism would write answers to questions through the skill of the demonstrator's ability to substitute his own written answer while feigning to blot the wet ink. (Photograph from Chapuis and Droz, 1958. Courtesy of Editions du Griffon, Neuchâtel, Switzerland)

Interfaces for Architecture Machines

Communication is the discriminatory response of an organism to a stimulus (Cherry, 1957). If we are to reckon with communication beyond formal rhetoric or syntax, whether English or computer graphics, we must address ourselves to the versatility of the discriminating mechanism—the interface. In this case the interface is the point of contact and interaction between a machine and the "information environment," most often the physical environment itself.

We have looked at graphic interfaces for one, and teletypes for another, but a dialogue demands a redundant and multichanneled concoction of sensory and motor devices far beyond these two mechanisms. We are talking about a total observation channel for an architecture machine.

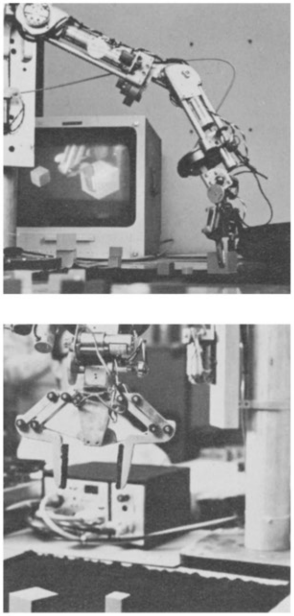
For a machine to have an image of a designer, of a problem, or of a physical environment, three properties are inherently necessary: an event, a manifestation, a representation. The event can be visual, auditory, olfactory, tactile, extrasensory, or a motor command. The manifestation measures the event with the appropriate parameters: luminance, frequency, brain wavelength, angle of rotation, and so forth. The representation is the act of mapping the information into a receptacle that is compatible with the organism's processing characteristics. These three properties—event, manifestation, representation—form the interface between any two organisms. The aspect of this interface with which we are primarily concerned is the manifestation, encompassed primarily by a piece of hardware.

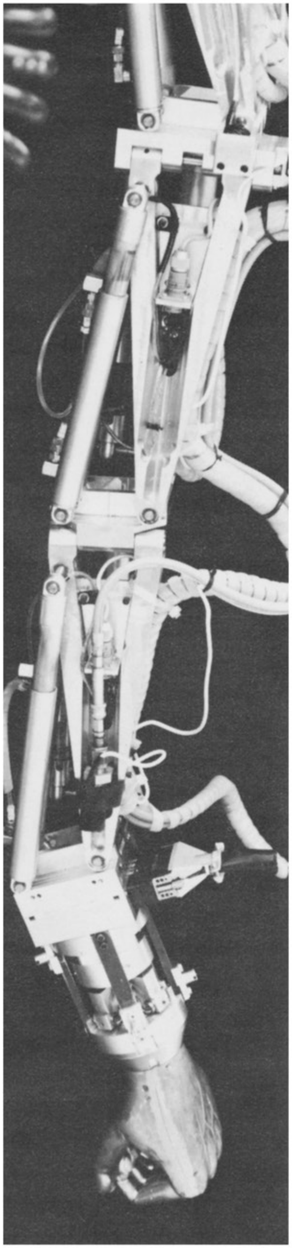
In an architect-machine relationship, perhaps the most interesting sensory interfaces are auditory and visual. Machines that are capable of visual perception and speech recognition are two of the prime targets of researchers in artificial intelligence. Someday, machines that can see and hear will be commonplace machines. Setting aside the phantasmagoria of robot designers, consider speaking to a machine that sees you—a machine with eyes and ears, a machine that walks and talks. In our present culture the thought is either frightening, foolish, or, to some, quite realistic. To our children it will be an ordinary daily occurrence. To Mortimer Taube (1961) it is offensive. To Marvin Minsky (1966) it is obvious.

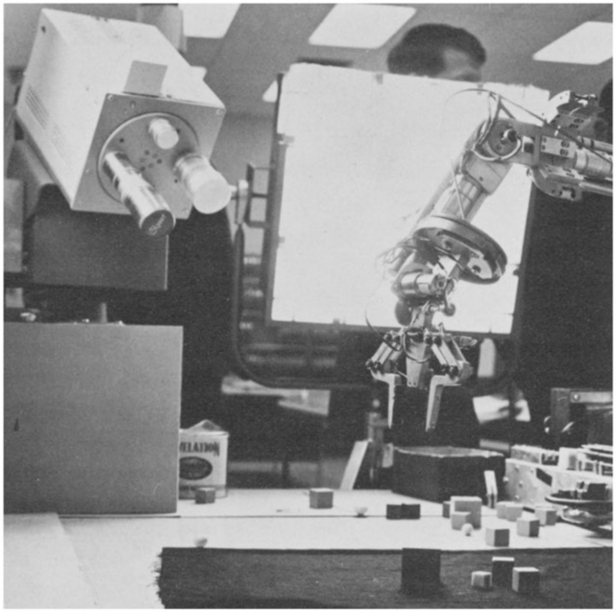
In the meantime, extrapolations into the future should recognize current problems of implementation. In the January 1967 issue of *Datamation,* Leslie Mezie portrayed a conversation with a machine that could listen, and talk. A fragment went as follows:

Professor:
What time is it?
Computer:
It is 8:30 p.m., Thursday, December 5, 1985.
Professor:
I think I would like to start with some music today, let's have some chamber music.
Computer:
You listen to Telemann most.
Professor:
No, something earlier.
Computer:
What about. . . .

Six months later, in the August issue of *Datamation,* B. W. Boehm parodied the sketch im-


1


2


3


4

1 The hand of "Butterfingers," the Stanford Hand-Eye Project. (J. A. Feldman et al., 1969)

2 The M.I.T. Robot Project's hand, Project MAC.

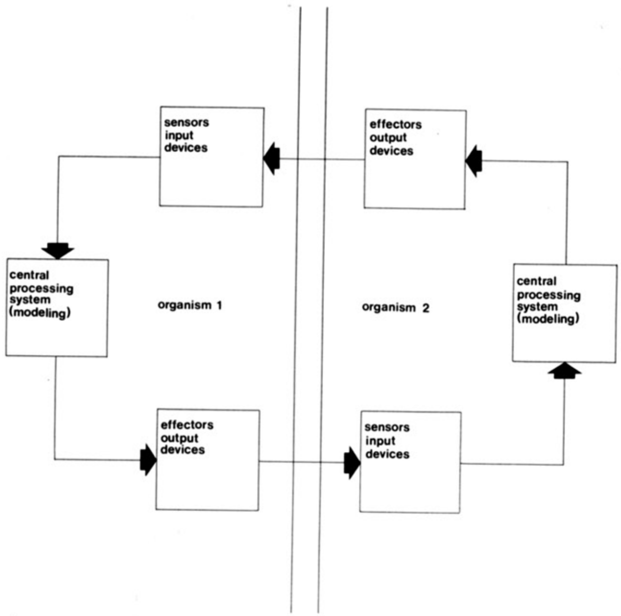
3 The eye and the hand of the Stanford project. This particular eye is a scanning device, a vidicon.

4 Voice input to the Stanford robot. Pierre Vicens of that project giving simple voice instructions (with a French accent), instructions like "pick up the big block."

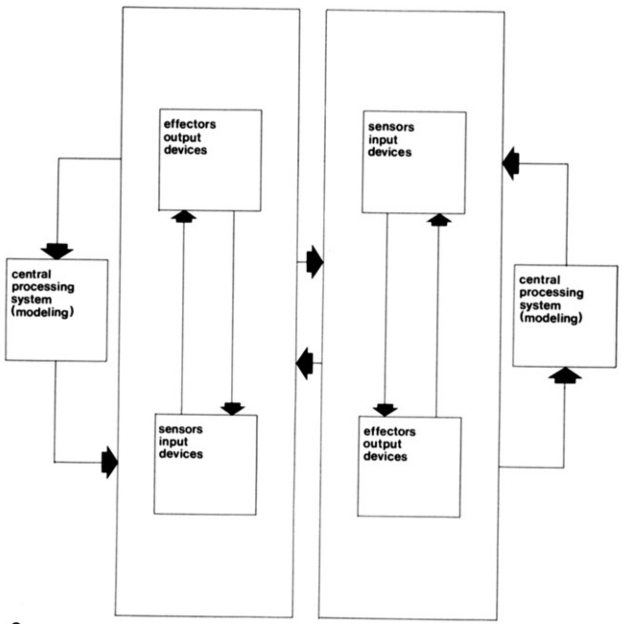
planting aspects of present-day dilemmas:
Professor:
What time is it?
Computer:
I did not catch the last word. Or was it two words?
Professor:
What is the time?
Computer:
The dime is a copper coin worth one-tenth of a dollar. The word derives from the Latin decem, meaning. . . .
Professor:
No. No. What is the time? The time?
Computer:
It is 8:30 p.m., Thursday, December 5, 1985. We have been having some trouble with your linguals recently. Sometimes I can't tell your d's from your t's. Let's practice them. Watch the display screen for the intonation pattern, and repeat after me: Teddy's daddy toted two dead toads to Detroit.
Professor:
Teddy's daddy toted. . . .

Nilo Lindgren's (1965a and b) comprehensive survey describes a host of intriguing research efforts in speech recognition, all of which fall into one of three catagories: the auditory sensation, the acoustical disturbance freely propagating through air, and a sequence of articulatory events in a psychological structure. The reader should also refer to the recent works of Bobrow and Klatt (1968), Reddy and Vicens (1968), and Rabiner (1968).

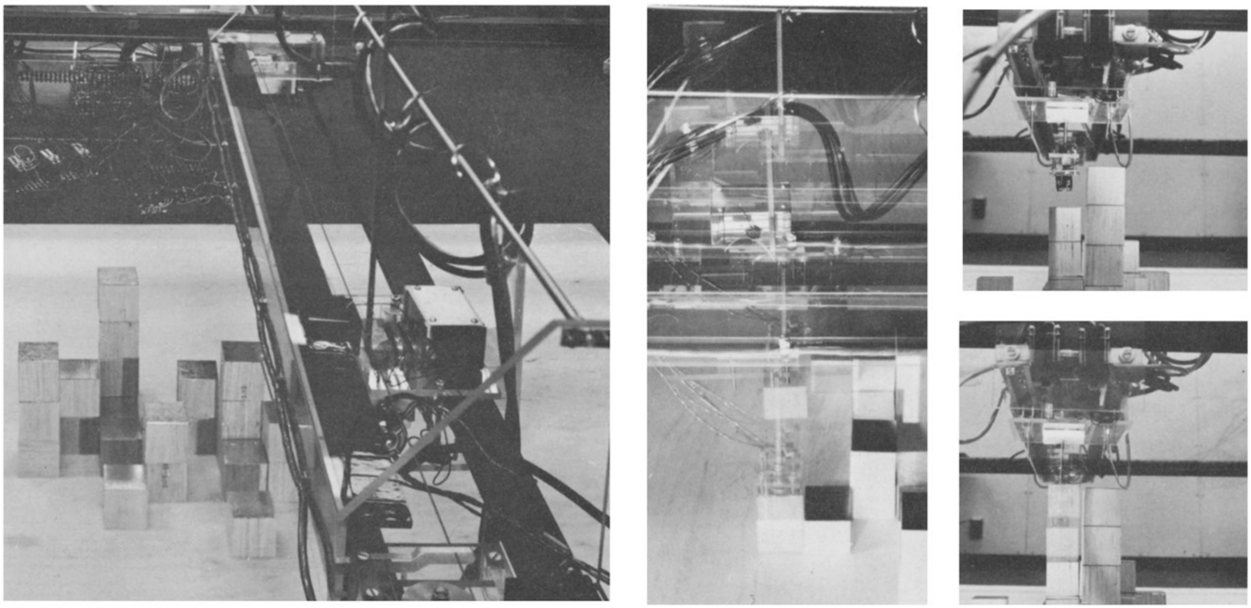
Beyond giving a machine ears, giving a machine eyes is extremely critical to architecture machines. Just on the hunch that a blind machine


sensors
input
devices
effectors
output
devices
central
processing
system
(modeling)
organism 1
organism 2
central
processing
system
(modeling)
effectors
output
devices
sensors
input
devices

1


effectors
output
devices
sensors
input
devices
central
processing
system
(modeling)
central
processing
system
(modeling)
sensors
input
devices
effectors
output
devices

2



3

1 The two diagrams represent an interface, in this case between man and machine. The left one is redrawn from Nilo Lindgren's "Human Factors in Engineering" (1966b). The important feature is that the "human factors" thinking treats the entire man-machine assemblage as a single entity. This implies that the interface is so smooth and so adaptable that in effect it does not exist.

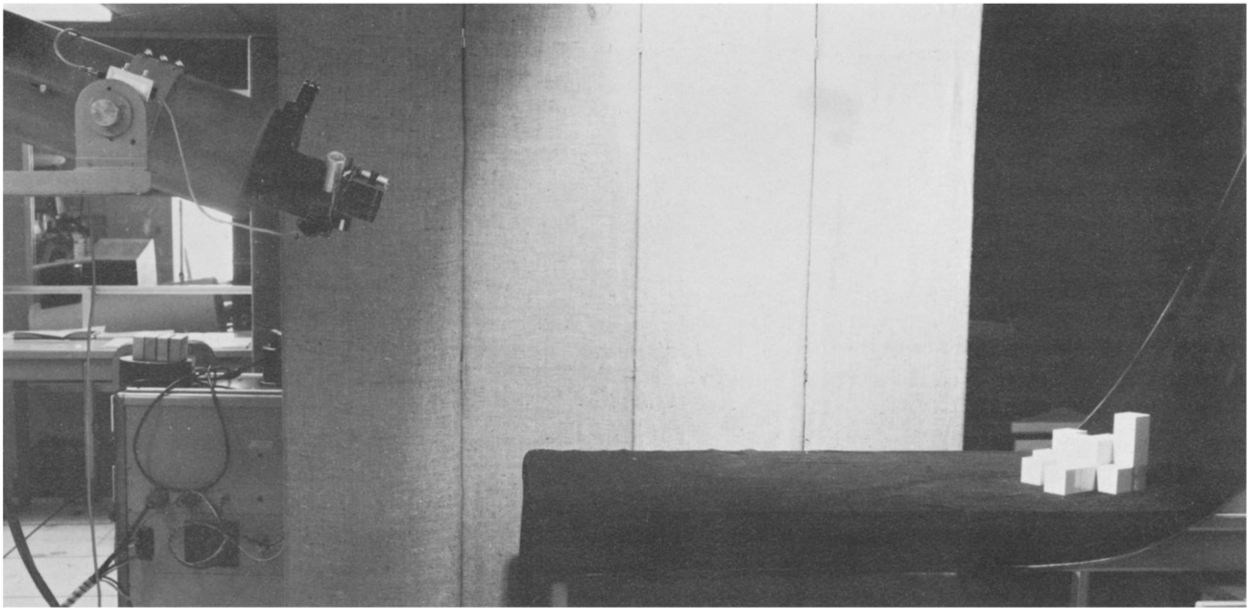
2 The illustration is redrawn from a reinterpretation of the above by Avery Johnson. Still considered as a single entity, the man-machine assemblage has a more active interface. In this interpretation, the interface has local computing power and can thus exhibit a behavior. This implies a continuous sensing and effecting mechanism, and it is the behavior of this device that is observed by both higher-order processors.

3 SEEK. This device is a homemade sensor/effector built by architecture students. The device has multiple attachments (magnets, photocells, markers, etc.) which it can position in three dimensions under computer control. It is anticipated that the mechanism will pile blocks, carry TV cameras, observe colors, and generally act as a peripheral device for student experiments in sensors and effectors that interact with the physical environment.

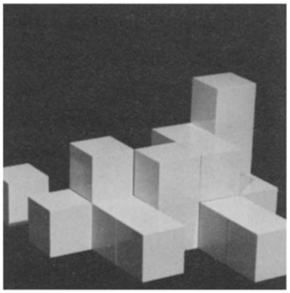
will have shortcomings similar to those of a blind architect, the relevance of a seeing machine warrants research. Outside of the design professions, giving machines eyes is of imminent importance. For instance, space exploration will eventually require machines that can both see and process the seen information. This is because the remote monitoring of a space robot's movements by earthlings requires too much transmission time (to Mars and back, for example), and a machine would crash into that which it is told to avoid only because the message to stop might arrive too late. More domestic applications involve visual discrimination of simple objects. Eventually, machines will package your purchased goods at the counter of your neighborhood supermarket.

Oliver Selfridge (and Neisser, 1963) is credited with the founding works in pattern recognition. His mechanism, PANDEMONIUM, would observe many localized visual characteristics. Each local verdict as to what was seen would be voiced by "demons" (thus, pandemonium), and with enough pieces of local evidence the pattern could be recognized. The more recent work of Marvin Minsky and Seymour Papert (1969) has extensively shown that solely local information is not enough; certain general observations are necessary in order to achieve complete visual discrimination.

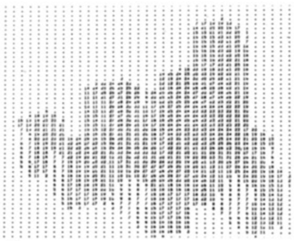
At present, these works are being applied to architectural problems as an exercise preliminary to the construction of an architecture machine. Anthony Platt and Mark Drazen are applying the Minsky-Papert eye to the problem of looking at physical models (Negroponte, 1969d). The interim goal of this



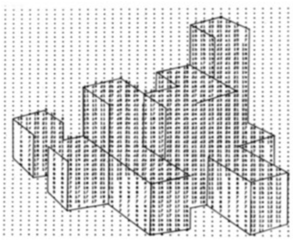
1



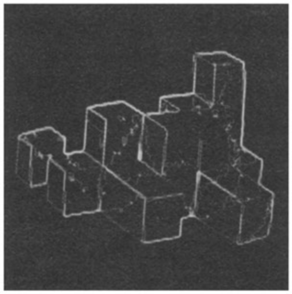
5



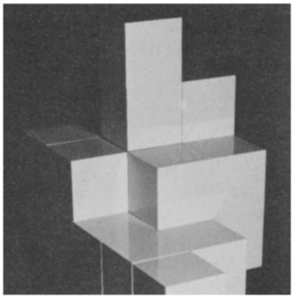
6



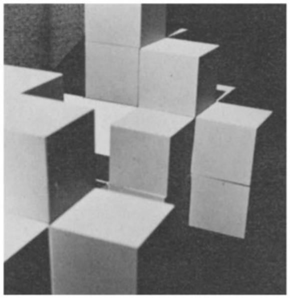
7



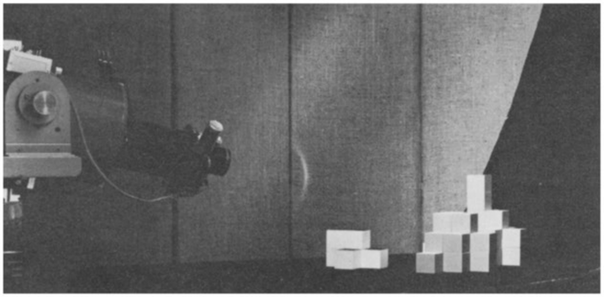
8



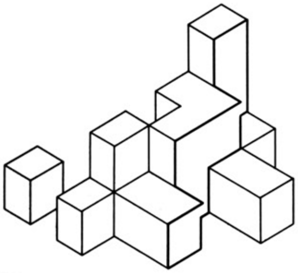
2



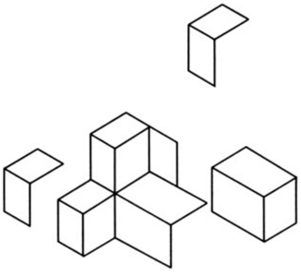
3



4



9



10

1 The M.I.T. Minsky/Papert eye. In this case the eye is an image dissector, a random-access device that does not scan back and forth but rather goes to discrete positions under computer control. This was the eye used for the Platt/Drazen vision experiment under the supervision of Seymour Papert.

2 Some vision problems — reflections and tone changes. Note that the top surfaces of the front lower cubes are a light gray, while the rear upper one has a black top surface. In other words, in such lighting the orientation of a surface cannot be assumed from its gray tone.

3 Other vision problems — depth of field and shadows.

4 More problems—disconnected bodies.

5 A typical model presented to the eye.

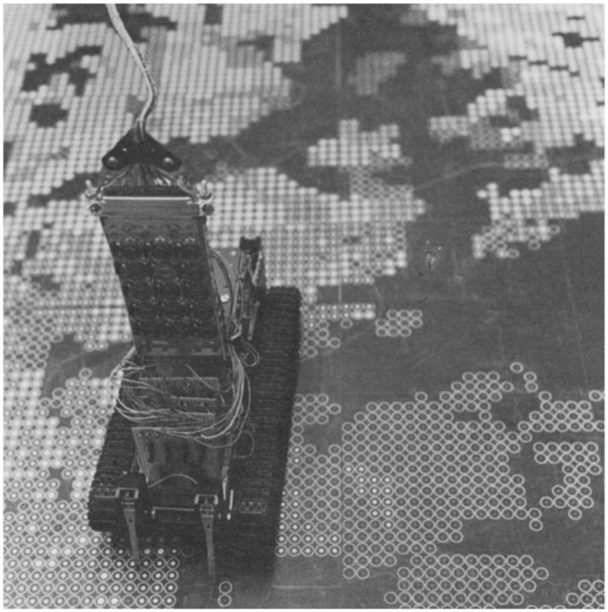
6 Printer output of the light intensities.

7 Contours of similar intensities.

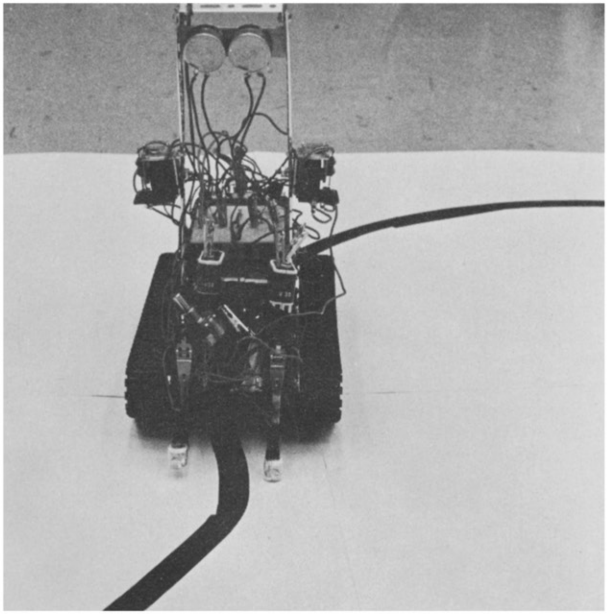
8 A cathode-ray tube display of the discovered contours. The "noise" is due to both bad lighting and a poor choice of contours.

9 The seen lines. These would be the lines seen under ideal, noise-free conditions.

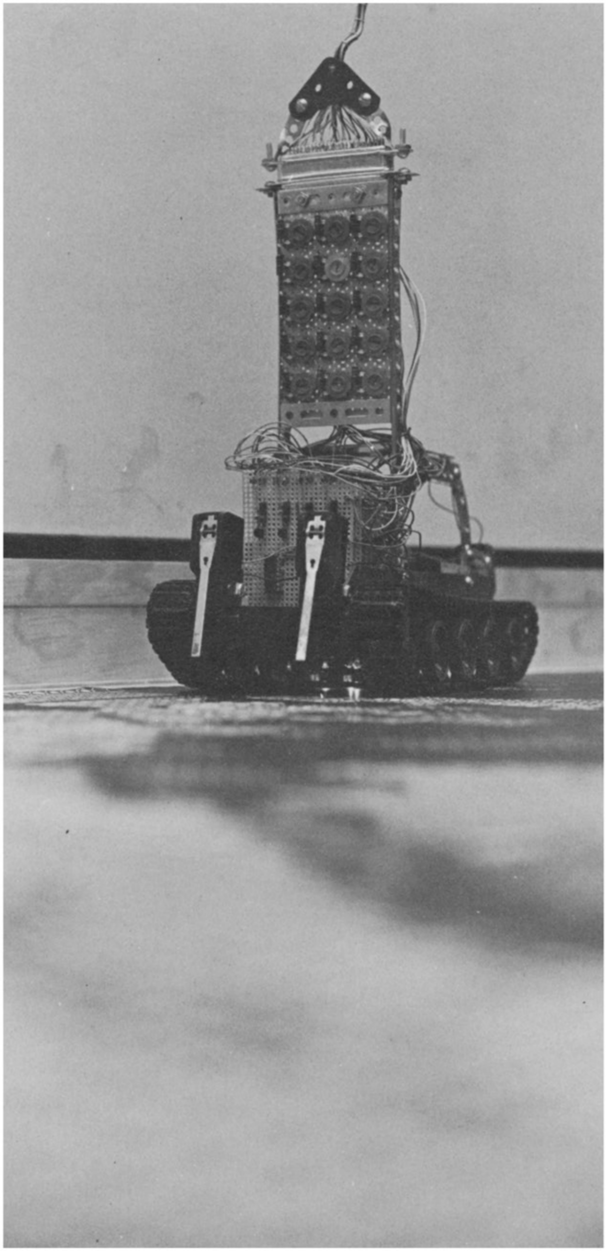
10 Minimal surfaces. A parallelogram indicates a complete surface that can be used as "strong evidence" to place others in space.



1



2



3

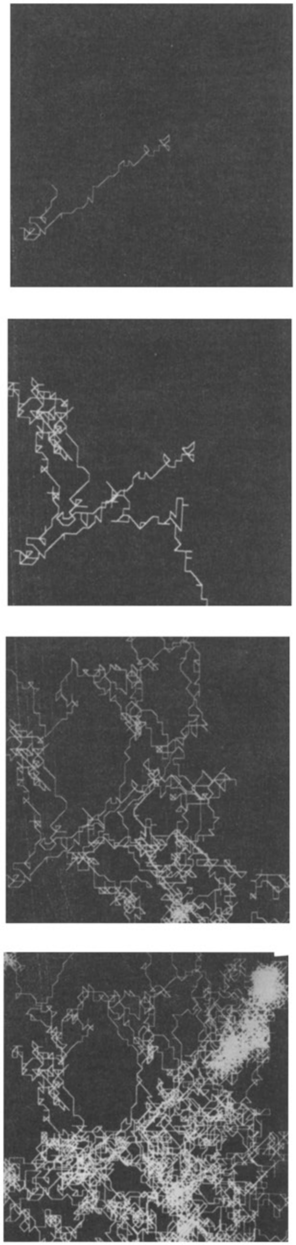
1 GROPE groping on the *Urban Atlas* map of New York's residential population density.

2 The old GROPE.

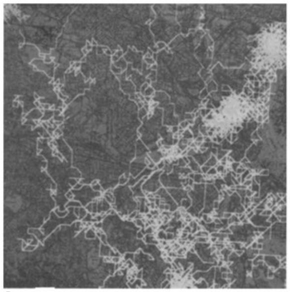
3 The new GROPE. The slight glow beneath GROPE is from three little lights that illuminate the area for the fifteen photocells. It is interesting to note that, like most Architecture Machine projects, GROPE started as a toy costing $15. Even though it has evolved into a major experiment, its circuitry and hardware have cost less than $80.

exercise is to observe, recognize, and determine the "intents" of several models built from plastic blocks. Combined with Platt's previously described LEARN, this experiment is an attempt at machine learning through machine seeing. In contrast to describing criteria and asking the machine to generate physical form, this exercise focuses on generating criteria from physical form.

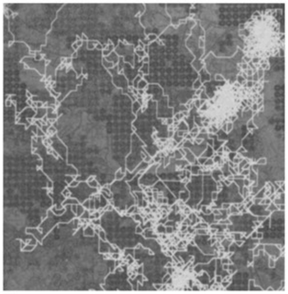
A second example of interfacing with the real world is Steven Gregory's GROPE (Negroponte et al., 1969b). GROPE is a small mobile unit that crawls over maps, in this case Passonneau and Wurman's (1966) *Urban Atlas* maps. It employs a low-resolution seeing mechanism constructed with simple photocells that register only states of on or off, "I see light" or "I don't see light." In contrast to the Platt experiment, GROPE knows nothing about images; it deploys a controller that must be furnished with a context and a role (as opposed to a goal: play chess as opposed to winning at chess). GROPE's role is to seek out "interesting things." To determine future moves, the little robot compares where he has been to where he is, compares the past to the present, and occasionally employs random numbers to avoid ruts. The onlooking human or architecture machine observes what is "interesting" by observing GROPE's behavior rather than by receiving the testimony that this or that is "interesting." At present, some aspects of GROPE are simulated and other aspects use the local computing power on GROPE's plastic back. GROPE will be one of the first appendages to an architecture machine, because it is an interface that explores the real world. An architecture machine must watch devices such as GROPE and observe



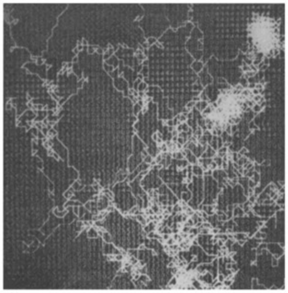
1



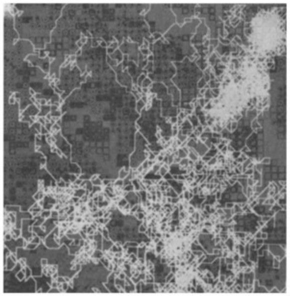
2



3



4



5

1 Before the Architecture Machine Project had its own dedicated computing power, aspects of GROPE were simulated on the ARDS display. The four illustrations represent a sequence that traces GROPE's path through an internal machine representation of *Urban Atlas* data for Boston. Note that by the fourth frame GROPE has "scrubbed out" two areas of the upper right. It turns out that this is Boston's downtown waterfront, indeed an "interesting" area of the map.

2 A photographic overlay of GROPE's path with a road map of Boston.

3 An overlay with "personal income" data.

4 An actual numerical display of the "personal income" data.

5 An overlay with "land use" and "residential density."

their behavior rather than listen to their comments.

But why not supply the machine with a coordinate description of the form on punch cards and proceed with the same experiment? Why must a machine actually see it? The answer is twofold. First, if the machine were supplied a nonvisual input, the machine could not learn to solicit such information without depending on humans. Second, it turns out that the computational task of simply seeing, the physiology of vision (as opposed to the psychology of perception) involves a set of heuristics that are apparently those very rules of thumb that were missing from LEARN, that made LEARN a mannerist rather than a student.

It seems natural that architecture machines would be superb clients for sophisticated sensors. Architecture itself demands a sensory involvement. Cardboard models and line drawings describe some of the physical and some of the visual worlds, but who has ever smelt a model, heard a model, lived in a model? Most surely, computer-aided architecture is the best client for "full interfacing." Designers need an involvement with the sensory aspects of our physical environments, and it is not difficult to imagine that their machine partners need a similar involvement.

Architecture Machines Learning Architecture

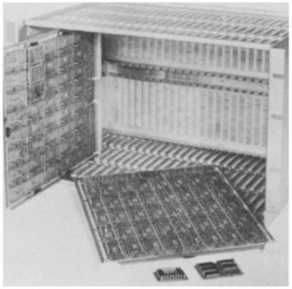
There is no security against the ultimate development of mechanical consciousness, in the fact of machines possessing little consciousness now . . . reflect upon the extraordinary advance which machines have made in the last few hundred years, and note how slowly the animal and the vegetable kingdoms are advancing.
Samuel Butler, *Erewhon*

When a designer supplies a machine with step-by-step instructions for solving a specific problem, the resulting solution is unquestionably attributed to the designer's ingenuity and labors. As soon as the designer furnishes the machine with instructions for finding a method of solution, the authorship of the results becomes ambiguous. Whenever a mechanism is equipped with a processor capable of finding a method "of finding a method of solution," the authorship of the answer probably belongs to the machine. If we extrapolate this argument, eventually the machine's creativity will be as separable from the designer's initiative as our designs and actions are from the pedagogy of our grandparents.

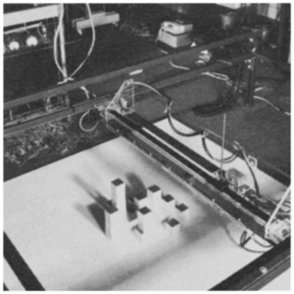
For a machine to learn, it must have the impetus to make self-improving changes, to associate courses with goals, to be able to sample for success and failure, and to be ethical. We do not have such machine capabilities; the problem is still theoretical, still of interest primarily to mathematicians and cyberneticians.

A 1943 theorem of McCulloch and Pitts states

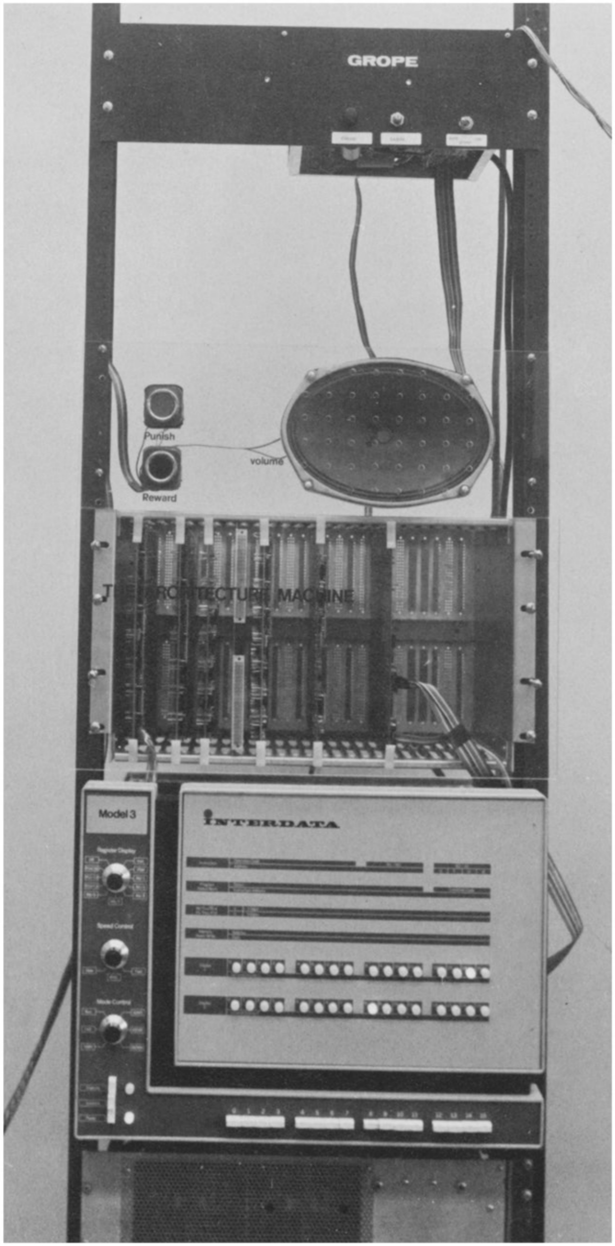

Model 3
INTERDATA
1


2


3


4


5


GROPE
Punish
Reward
volume
Model 3
INTERDATA
6

1 The Interdata computer, at present the nucleus of the Architecture Machine Project. (Photograph courtesy of the Interdata Corporation)

2 Interdata's mother-boards and daughter-boards. These panels hold the circuitry and plug into a chassis. This simplicity permits rapid interfacing of peripheral sensors and effectors. (Photograph courtesy of the Interdata Corporation)

3 The Architecture Machine's first technician.

4 SEEK and its controller.

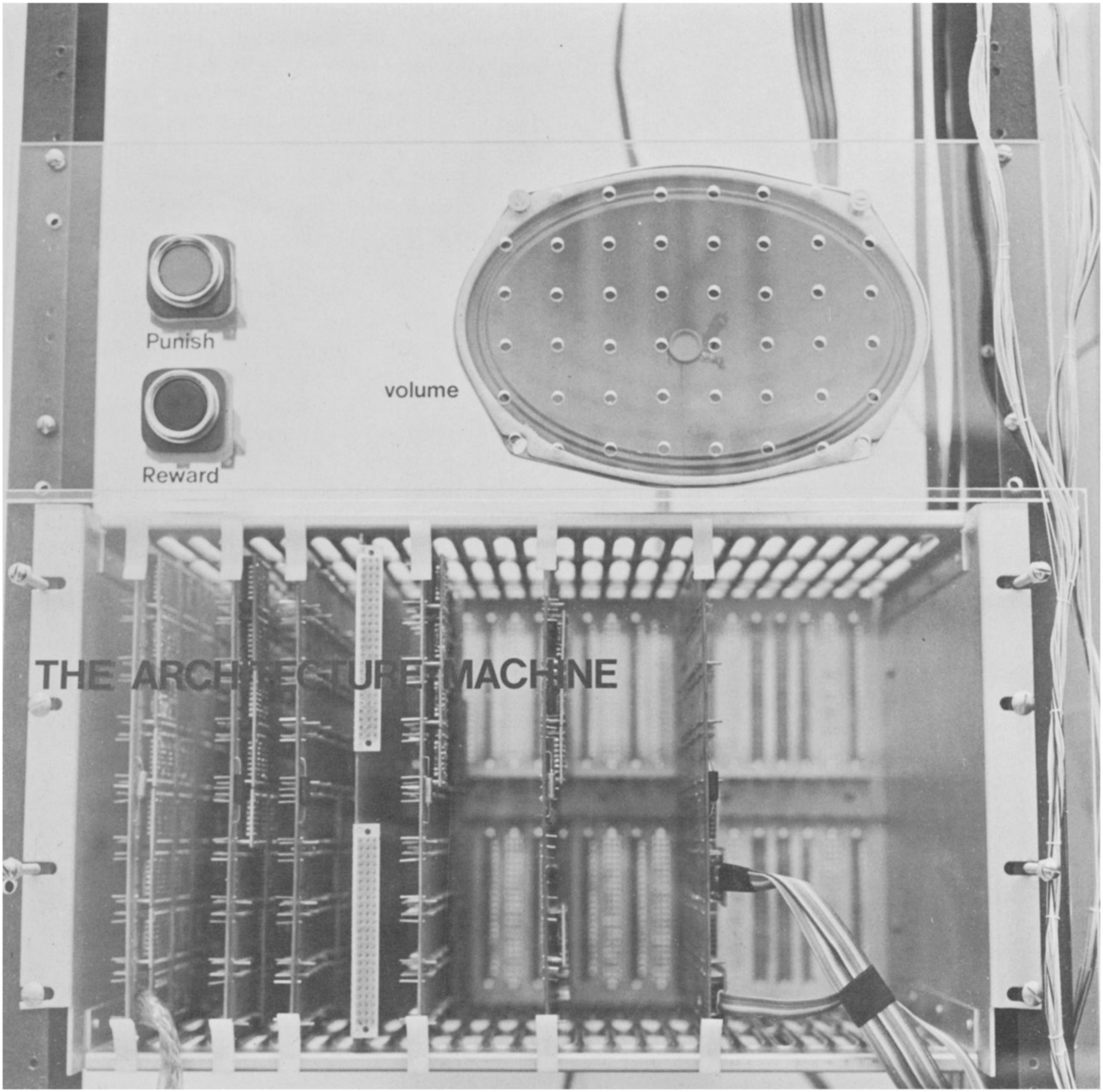
5 The Architecture Machine configuration including the three ARDS. (September 1, 1969)

6 Processor, expansion chassis, and various controllers.

that a machine constructed with regenerative loops of a certain formal character is capable of deducing any legitimate conclusion from a finite set of premises. One approach to such a faculty is to increase the probability of meaningfulness of the output (the design) generated from random or disorderly input (the criteria). Ross Ashby (1956) states, "It has been often remarked that any random sequence, if long enough, will contain all answers; nothing prevents a child from doodling: $cos^2X + sin^2X = 1$." In the same spirit, to paraphrase the British Museum/chimpanzee argument, a group of monkeys, while randomly doodling, can draw plans, sections, and elevations of all the great works of architecture and do this in a finite period of time. As the limiting case, we would have a tabula rasa, realized as a network of uncommitted design components or uncommitted primates. Unfortunately, in this process our protagonists will have built Levittown, Lincoln Center, and the New York Port Authority Towers.

Surely some constraint and discrimination is necessary if components are to converge on solutions within "reasonable" time. Components must assume some original commitment. As examples of such commitment, five particular subassemblies should be part of an architecture machine: (1) a heuristic mechanism, (2) a rote apparatus, (3) a conditioning device, (4) a reward selector, and (5) a forgetting convenience.

A heuristic is a method based on rules of thumb or strategies that drastically limit the search for a solution. A heuristic method does not guarantee a solution, let alone an optimal


Punish
Reward
volume
THE ARCHITECTURE MACHINE

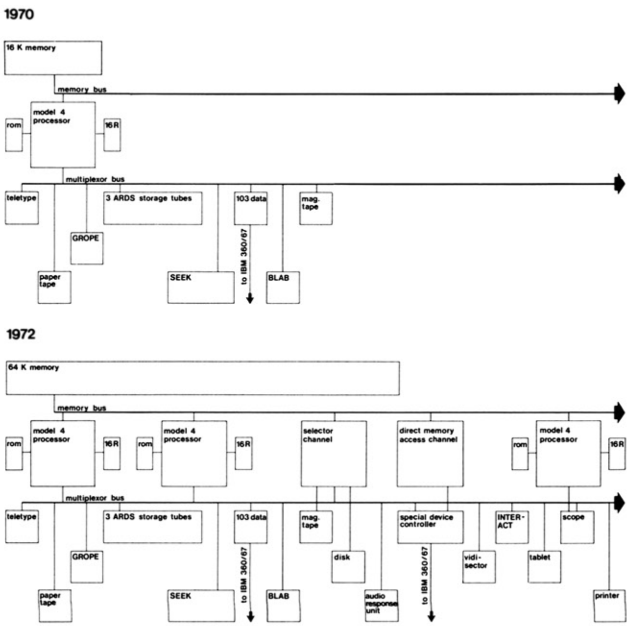
1 The Architecture Machine's punish/reward and BLAB, a rudimentary audio output device.

one. The payoff is in time and in the reduction of the search for alternatives. Heuristic learning is particularly relevant to evolutionary machines because it lends itself to personalization and change by talking to one specific designer, overviewing many designers, or viewing the real world. In an architecture machine, this heuristic element would probably be void of specific commitment when the package arrives at an office. Through architect-sponsored maturation, a resident mechanism would acquire broad rules to handle exceptional information. The first time a problem is encountered, the machine would attempt to apply procedures relevant to similar problems or contexts. Heuristics gained from analogous situations would be the machine's first source of contribution to the solution of a new problem.

After repeated encounters, a rote apparatus would take charge. Rote learning is the elementary storing of an event or a basic part of an event and associating it with a response. When a situation is repeatedly encountered, a rote mechanism can retain the circumstance for usage when similar events are next encountered. In architecture, this repetition of subproblems is extremely frequent: parking, elevators, plumbing, and so forth. And again a rote mechanism lends itself to evolutionary expansion. But, unlike a heuristic mechanism, this device would probably come with a small original repertoire of situations it can readily handle.

Eventually, simple repetitous responses become habits, some good and some bad. More specifically acclimatized than a rote apparatus,

1



2

1 Diagram taken from Marvin Minsky and Seymour Papert's *Perceptrons* (1969).

2 Projections of the present Architecture Machine configuration.

a conditioning mechanism is an enforcement device that handles all the nonexceptional information. Habits, not thought, assist humans to surmount daily obstacles. Similarly, in a machine, beyond rote learning, design habitudes can respond to standard events while the designer, the heuristic mechanism, and the rote apparatus engage in the problem-solving and problem-worrying (Anderson, 1966) aspects of design. Each robot would develop its own conditioned reflexes (Uttley, 1956). Like Pavlov's dog, the presence of habitual events will trigger predefined responses with little effort until the prediction fails; whereupon, the response is faded out by frustration (evolution) and is handled elsewhere in the system.

A reward selector initiates no activities. In a Skinnerian fashion (B. F. Skinner, 1953), the reward mechanism selects from any action that which the "teacher" likes. The teachers (the designer, the overviewing apparatus, the inhabitants) must exhibit happiness or disappointment for the reward mechanism to operate. Or, to furnish this mechanism with direction, simulation techniques must evolve that implicitly pretest any environment. The design of this device is crucial; bad architecture could escalate as easily as good design. A reward selector must not make a machine the minion or bootlicker of bad architecture. It must evaluate, or at least observe, goals as well as results.

Finally, unlearning is as important as learning (Brodey, 1969c). The idea of "its [the computer's] inability to forget anything that has been put into it. . . ." (A. Miller, 1967) is simply fallacious. Information can assume less significance over time and eventually disappear—exponential forgetting. Obsolescence can occur through time or pertinence. A technological innovation in the construction industry, for example, can make entire bodies of knowledge obsolete (which, as humans, we tend to hate surrendering). Or past procedures might not satisfy environmental conditions that have changed over time, thus invalidating a heuristic, rote response, or conditioned reflex.

These five items are only pieces of an architecture machine; the entire body will be an ever-changing group of mechanisms that will undergo structural mutations, bear offspring (Fogel et al., 1965), and evolve, all under the direction of a cybernetic device.

Robot Architects

Rather than "problem-solving," I characterized the design process as "problem-worrying." I suggested that architecture is concerned with structuring man's environment to facilitate the achievement of human purposes (intellectual, psychological and utilitarian) where those purposes are incompletely known and cannot be extrapolated from what is given in the situation. Rather, human purposes are altered by the very environment that is created to facilitate them. The structuring of the environment must be accomplished, then, through the exercise of tentative foresight and the critical examination of that foresight and the actions to which it leads. According to this description, neither the human purposes nor the architect's methods are fully known in advance. Consequently, if this interpretation of the architectural problem situation is accepted, any problem-solving technique that relies on explicit problem definition, on distinct goal-orientation, on data collection, or even on non-adaptive algorithms will distort the design process and the human purposes involved.
Stanford Anderson, "Problem-Solving and Problem-Worrying"

It is interesting to ponder what a human designer must do or the behavior he must exhibit in order to be a good architect, a talented architect, an ethical architect—not, perforce, a successful architect. We know that he must somehow contribute and promote physical environments that both house and stimulate the good life. But we do not know much about the good life; it has no "utility function" and cannot be optimized. We know that he must have an understanding of and ease with physical form. But we do not know how our own cognitive processes visualize shape and geometry. We know that he must interpret human needs and desires. But we do not know how to acquire these needs and desires.

What probably distinguishes a talented, competent designer is his ability both to provide and to provide for missing information. Any environmental design task is characterized by an astounding amount of unavailable or indeterminate information. Part of the design process is, in effect, the procurement of this information. Some is gathered by doing research in the preliminary design stages. Some is obtained through experience, overlaying and applying a seasoned wisdom. Other chunks of information are gained through prediction, induction, and guesswork. Finally some information is handled randomly, playfully, whimsically, personally.

It is reasonable to assume that the presence of machines, of automation in general, will provide for some of the omitted and difficult-to-acquire information. However, it would appear foolish to suppose that, when machines know how to design, there will be no missing information or that a single designer can give the machine all that it needs. Consequently, we, the Architecture Machine Group at M.I.T., are embarking on the construction of a machine that can work with missing information. To do this, an architecture machine must understand our metaphors, must solicit information on its own, must acquire experiences, must talk to a wide variety of people, must

1

2

1 The Jaquet-Droz Writer (circa 1774). The little boy is 28 inches tall, carved from wood, and composed of a very complicated mechanism which still works. It has carefully written, "I do not think, therefore I will never be." (Photograph courtesy of Editions du Griffon, automaton in Neuchâtel Museum)

2 "And it will serve us right" (Asimov, 1969). (Reprinted from "Psychology Today," magazine, April 1969, © Communications/Research/Machines, Inc. Photograph by Stephen Wells) improve over time, and must be intelligent. It must recognize context, particularly changes in goals and changes in meaning brought about by changes in context.

In contrast, consider for a moment a society of designers built upon machine aides that cannot evolve, self-improve, and most importantly, cannot discern shifts in context. These machines would do only the dull ignoble tasks, and they would do these tasks employing only the procedures and the information designers explicitly give them. These devices, for example, could indiscriminately optimize partial information and generate simplistic solutions that minimize conflicts among irrelevant criteria. Furthermore, since no learning is permitted in our not-so-hypothetical situation, these machines would have the built-in prejudices and "default options" of their creators. These would be unethical robots.

Unfortunately most researchers seem to be opting for this condition. As a result, many computer-aided design studies are relevant only insofar as they present more fashionable and faster ways of doing what designers already do. And since what designers already do does not seem to work, we will get inbred methods of work that will make bad architecture, unresponsive architecture, even more prolific.

I therefore propose that we, architects and computer scientists, take advantage of the professional iconoclasms that exist in our day —a day of evolutionary revolution; that we build machines equipped with at least those devices that humans employ to design. Let us build machines that can learn, can grope, and can fumble, machines that will be architectural partners, architecture machines.

Bibliography

Aguilar, R. J., and J. E. Hand	A Generalized Linear Model for Optimization of Architectural Planning	American Federation of Information Processing Proceedings, Spring Joint Computer Conference, 32, 81-88	1968
Ahuja, D. V., and S. A. Coons	Geometry for Construction and Display	IBM Systems Journal, 7, Nos. 3 and 4, 188-205	1968
Ainsworth, W. A.	On the Efficiency of Learning Machines	Institute of Electrical and Electronics Engineers Transactions, Systems Science and Cybernetics, SSC-3, 77-85	1967 November
Alexander, C.	Systems Generating Systems	Systemat, 1, 1-11	1967
	The Question of Computers in Design	Landscape, 14, No. 3, 6-8	1965 Spring
	Notes on the Synthesis of Form	Cambridge, Mass.: Harvard University Press	1964
	A Much Asked Question about Computers and Design	Architecture and the Computer. Boston: First Boston Architectural Center Conference, 52-56	1964 December 5
Alexander, C., S. Ishikawa, and M. Silverstein	A Pattern Language Which Generates Multi-Service Centers	Berkeley, Calif.: Center for Environmental Studies	1968
Amarel, S.	On the Mechanization of Creative Processes	Institute of Electrical and Electronics Engineers Spectrum, 3, No. 4, 112-114	1966 April
Amari, S.	A Theory of Adaptive Pattern Classifiers	Institute of Electrical and Electronics Engineers Transactions on Electric Composition, EC-16, 299-307	1967
Anderson, A. R. (editor)	Minds and Machines	Englewood Cliffs, N.J.: Prentice-Hall	1964
Anderson, S.	Commentary	Architectural Design, 39, 514	1969 September

Anderson, S.	Problem Solving and Problem Worrying	American Institute of Architects Teachers Seminar, Bloomfield, Mich.: Cranbrook Academy	1966 June
Anderson, S. (editor)	Planning for Diversity and Choice	Cambridge, Mass.: The M.I.T. Press	1968
Ashby, W. R.	The Brain of Yesterday and Today	1967 Institute of Electrical and Electronics Engineers, International Convention Record, Part 9, 30-33	1967
	The Design of an Intelligence Amplifier	Automata Studies, C. E. Shannon and J. McCarthy (editors). Princeton: Princeton University Press	1956
Asimov, I.	And It Will Serve Us Right	Psychology Today, 2, No. 11, 38-41	1969 April
	The Perfect Machine	Science Journal, 4, No. 10, 115-118	1968 October
	The Rest of the Robots	New York: Doubleday	1964
	Robbie	Of Men and Machines. New York: Dutton, 140-158	1963
	I, Robot	New York: Doubleday	1950
Barnett, J.	Computer-Aided Building Design; Where Do We Go From Here?	Architectural Record, 141, 219-220	1967 April
	Computerized Cost Estimating	Architectural Record, 141, 163-166	1967 March
	Computer Revolution: How Does It Affect Architecture?	Architectural Record. 140, 168-170	1966 July
	Computer-Aided Design and Automated Working Drawings	Architectural Record, 138, 85-98	1965 October
	Will the Computer Change the Practice of Architecture?	Architectural Record, 137, 143-150	1965 January

Barron, R. L.	Self-Organizing and Learning Control Systems	Cybernetic Problems in Bionics, H. L. Oestreicher and D. R. Moore (editors). New York: Gordon and Breach	1968
Bazilevskii, Y. (editor)	The Theory of Mathematical Machines	New York: Pergamon Press, Macmillan	1963
Becker, R., and F. Poza	Natural Speech from a Computer	Proceedings of 23rd Association for Computing Machinery National Conference, 795-800	1968
Beer, S.	Machines that Control Machines	Science Journal, 4, No. 10, 89-96	1968 October
	Cybernetics Thrills and Threats	The Computer Bulletin, 11, No. 4, 305-307	1968 March
Berkeley, Edmund	LIES — Lying Invalidates Excellent Systems	Computers and Automation, 16, No. 1, 7	1967 January
Berkeley, Ellen	Computers for Design and A Design for the Computer	The Architectural Forum, 128, No. 2, 60-65	1968 March
Bernholtz, A.	RUMOR, The Random Generation and Evaluation of Plans	Laboratory for Computer Graphics and Spatial Analysis, Graduate School of Design, Cambridge, Mass.: Harvard University	1969
	Some Thoughts on Computers, Role Playing and Design	Connection, 5, Nos. 2 and 3, 88-91	1968 Winter-Spring
	Computer-Graphic Displays in Architecture	Journal of the Society for Information Display, 3, No. 2, 52-55	1966 March/April
	Design and the Computer	Toronto, Canada: Southam Business Publications	1965
Bernholtz, A., and E. Bierstone	Computer-Augmented Design	Design Quarterly 66/67, 41-52	1966 December
	Computer Augmented Design	Design and Planning 2, M. Krampen and P. Seitz (editors). New York: Hastings House, 41-52	1966

Bernstein, A., and H. Rubin	Artificial Evolution of Problem-Solvers	The American Behavioral Scientist, 8, No. 9, 19-23	1965 May
Beshers, J. M.	Population Processes in Social Systems	New York: Free Press	1967
Beshers, J. M. (editor)	Computer Methods in the Analysis of Large Scale Social Systems	Cambridge, Mass.: Joint Center for Urban Studies	1965
Bijl, B., and A. Renshaw	Application of Computer Graphics to Architectural Practice	Edinburgh: Architecture Research Unit, University of Edinburgh	1968
Birmingham, H. P.	Human Factors in Electronics — Historical Sketch	Institute of Radio Engineers, Proceedings, 50, 1116-1117	1962 May
Bishop, M.	The Reading Machine	Of Men and Machines, A. O. Lewis, Jr. (editor). New York: Dutton, 315-317	1963
Blasi, C., R. Galimberti, G. Padovano, and I. DeLotto	A Program for Computer-Aided Urban Planning	Calcolo, 5, 229-249	1968 April-June
Block, H. D.	Analysis of Perceptrons	Proceedings of the Western Joint Computer Conference, 281-289	1961
Bobrow, D. G.	Natural Language Input for a Computer Problem-Solving System	Cambridge, Mass.: The M.I.T. Department of Mathematics	1964
	Syntactic Analysis of English by Computer — A Survey	American Federation of Information Processing Proceedings, Fall Joint Computer Conference, 24, 365-387	1963
Bobrow, D. G. and D. H. Klatt	A Limited Speech Recognition System	American Federation of Information Processing Proceedings, Fall Joint Computer Conference, 33, 305-318	1968
Boehm, B. W.	The Professor and the Computer: 1985	Datamation, 13, No. 8, 56-58	1967 August
Boguslaw, R.	The New Utopians	Englewood Cliffs, N. J.: Prentice-Hall	1965

Boorn, J., S. Hoderowski, and A. Fleisher	CHOICE	Architectural Design, 39, 510	1969 September
Bowden, B. V. (editor)	Faster than Thought	New York: Pitman	1953
Bowles, E. A.	Computers in Humanistic Research	Englewood Cliffs, N.J.: Prentice-Hall	1967
	The Humanities and the Computer: Some Current Research Problems	Computers and Automation, 15, No. 4, 24-27	1966 April 15
Brodey, W. M.	Information Exchange Modelled in the Time Domain	General Systems Theory and Psychiatry, W. Gray, F. A. Duhl, and N. B. Rizzo (editors). Boston: Little, Brown	1969a (in press)
	Experiments in Evolutionary Environmental Ecology	Computer Graphics in Architecture and Design, M. Milne (editor). New Haven: Yale School of Art and Architecture	1969b
	Unlearning the Obsolescent	Architectural Design, 39, 483-484	1969c September
	The Clock Manifesto	Annals of the New York Academy of Science. 138, 895-899	1967
	Soft Architecture—The Design of Intelligent Environments	Landscape, 17, No. 1, 8-12	1967 Autumn
	Unlearning the Obsolescent	Institute of Electrical and Electronics Engineers, Systems, Science and Cybernetics Conference	1966 (in press)
	Developmental Learning and the Education of the Child Born Blind	Etc., A review of General Semantics, 23, 293-306	1965 September
	Sound and Space	American Institute of Architects Journal, 42, No. 1, 58-60	1964 July

Brodey, W. M., and A. R. Johnson	A Visual Prosthesis that Looks	Association of Computing Machinery's Second Conference on Visual Prosthesis, Chicago	1969
Brodey, W. M., and N. Lindgren	Human Enhancement: Beyond the Machine Age	Institute of Electrical and Electronics Engineers Spectrum, 5, No. 2, 79-93	1968 February
	Human Enhancement Through Evolutionary Technology	Institute of Electrical and Electronics Engineers Spectrum, 4, No. 9, 87-97	1967 September
Butler, S.	Erewhon	New York: Dutton	1923
Campion, D.	Computers in Architectural Design	London: Elsevier	1968
	Design Simulation by Computer	Architectural Review, 140, 460	1966 December
Capek, K.	Rossum's Universal Robots	London: Oxford University Press	1961
Chapuis, A., and E. Droz	Automata	Neuchâtel, Switzerland: Editions du Griffon	1958
Chasen, S. H.	The Introduction of Man-Computer Graphics into the Aerospace Industry	American Federation of Information Processing Proceedings, Fall Joint Computer Conference, 27, 883-892	1965
Cherry, C.	On Human Communication	Cambridge, Mass.: The M.I.T. Press	1957
Cherry, C. (editor)	Information Theory	New York: Butterworth	1961
Childs, D. L.	Description of a Set-Theoretic Data Structure	Technical Report 3. Ann Arbor: University of Michigan	1968
Clark, W. E., and J. J. Souder	Man and Computer in the Planning Process	Architecture and the Computer. Boston: First Boston Architectural Center Conference, 29-33	1964 December 5
Cogswell, A. R.	Housing, the Computer, and the Architectural Process	Law and Contemporary Problems, 32, No. 2, 274-285	1967 Spring
Cohen, J.	Human Robots in Myth and Science	London: Allen & Unwin	1966

Coles, L. S.	Talking with a Robot in English	Proceedings of the International Joint Conference on Artificial Intelligence, 587-596	1969 May 7-9
	An On-Line Question-Answering System with Natural Language and Pictorial Input	Proceedings, Association of Computing Machinery 23rd National Conference, 157-167	1968 November
Comba, P. G.	A Language for Three-Dimensional Geometry	IBM Systems Journal, 7, Nos. 3 and 4, 292-308	1968
Cook, P.	Architecture: Action and Plan	London: Reinhold	1967
Coons, S. A.	Surfaces for Computer-Aided Design of Space Forms	Cambridge, Mass.: Project MAC, M.I.T.	1967 June
	Computer-Aided Design	Design Quarterly 66/67, 7-14	1966 December
	The Uses of Computers in Technology	Scientific American, 215, 177-188	1966 September
	Computer Graphics and Innovative Engineering Design	Datamation, 12, No. 5, 32-36	1966 May
	Computer-Aided Design	Architecture and the Computer. Boston: First Boston Architectural Conference, 26-28	1964 December 5
	An Outline of the Requirements for Computer-Aided Design Systems	Cambridge, Mass.: Electronic Systems Laboratory Technical Memorandum 169, M.I.T.	1963 March
Cote, A. J., Jr.	The Search for the Robots	New York: Basic Books	1967
Courtieux, G.	Récents Développements dans les Applications des Ordinateurs a la Construction Civile, a l'Architecture et a l'Urbanisme	L'Ingénieur Constructeur, 47-57	1968 July-August
David, E. E., Jr.	Digital Simulation in Research on Human Communication	Institute of Radio Engineers, Proceedings, 49, No. 1, 319-329	1961 January

Davis, R. H.	The Computer is Neutral	Computer Yearbook and Directory, First Edition. Detroit: American Data Processing, 29-39	1966
Donnelly, T. G., F. S. Chapin, Jr., and S. F. Weiss	A Probabilistic Model for Residential Growth	Chapel Hill: Institute for Research in Social Science, University of North Carolina	1964
Dreyfus, A. L.	Why Computers Must Have Bodies in Order to Be Intelligent	Review of Metaphysics, 21, No. 1, 13-32	1967 September
	Alchemy and Artificial Intelligence	IBM Corporation AD-625719, 95	1965 December
Duda, R. O., and P. E. Hart	Graphical-Data-Processing Research Study and Experimental Investigation	California: Stanford Research Institute, AD-673719	1968 August
Duhl, L. J.	Planning and Predicting: Or What to Do When You Don't Know the Names of the Variables	Daedalus, 96, No. 3, 779-788	1967 September
Eastman, C. M.	Cognitive Processes and Ill-Defined Problems: A Case Study from Design	Proceedings of the International Joint Conference on Artificial Intelligence, 655-668	1969
Eberhard, J. P.	The City as a System	Beyond Left and Right, R. Kostelanetz (editor). New York: Morrow	1968a
	A Humanist Case for the Systems Approach	American Institute of Architects Journal, 50, No. 1, 34-38	1968b July
Eden, M.	Human Information Processing	Institute of Electrical and Electronics Engineers, Transactions on Information Theory, IT-9, 253-256	1963 October
Eisenberg, L.	What Computers Can't Do	Harpers, 231, 96-99	1965 August
Emery, J. C.	The Planning Process and Its Formulation in Computer Models	Proceedings of the Second Congress of Information System Sciences, 369-389	1964
Englebart, D. C.	Augmenting Human Intellect: A Conceptual Framework	Stanford Research Institute Report, AD289565	1962 October

Englebart, D. C., and W. K. English	A Research Center for Augmenting Human Intellect	American Federation of Information Processing Proceedings, Fall Joint Computer Conference, 33, 395-410	1968
Evans, D., C. Wylie, G. Romeny, and A. Erdahl	Half-tone Perspective Drawings by Computer	American Federation of Information Processing Proceedings, Fall Joint Computer Conference, 31, 46-58	1967
Evans, G. W., G. F. Wallace, and G. L. Sutherland	Simulation Using Digital Computers	Englewood Cliffs, N.J.: Prentice-Hall,	1967
Ewald, W. R., Jr. (editor)	Environment and Policy	Bloomington: Indiana University Press	1968
Fair, G. R., A. D. J. Flowerdew, W. G. Munro, and D. Rowley	Note on the Computer as an Aid to the Architect	Computer Journal, 9, No. 1, 16-20	1966 May
Fano, R. M.	The Computer Utility and the Community	Institute of Electrical and Electronics Engineers International Convention Record, Part 12, 30-36	1967
Feigenbaum, E. A., and J. Feldman (editors)	Computers and Thought	New York: McGraw-Hill	1963
Feldman, J., G. Feldman, G. Falk, G. Grape, J. Pearlman, I. Sobel, and J. Tennebaum	The Stanford Hand-Eye Project	Proceedings of the International Joint Conference on Artificial Intelligence, 509-520	1969
Feldt, A. G.	Operational Gaming in Planning and Architecture	Prepared for presentation at the American Institute of Architects-Researchers Conference, Gatlinburg, Tenn.	1967 October 25
	Operational Gaming in Planning Education	American Institute of Planners Journal, 32, No. 1, 17-23	1966 January
Fenves, S. J.	Computer Use in Building Design	Building Research, 3, No. 2, 10-12	1966 March-April
Ferry, W. H.	Must We Rewrite the Constitution to Control Technology	Saturday Review, 51, No. 9, 50-54	1968 March 2

Fetter, W. A.	Computer Graphics	Design Quarterly 66/67, 15-24	1966 December
	Computer Graphics in Communications	New York: McGraw-Hill	1964a
	Computer Graphics: Architecture and the Computer	Boston: First Boston Architectural Center Conference, December 5, 34-36	1964b
Fleisher, A., W. Porter, and K. Lloyd	DISCOURSE: Computer Assisted City Design	Computer Graphics in Architecture and Design. M. Milne (editor). New Haven: Yale School of Art and Architecture	1969
Fogel, L. J.	On the Design of Conscious Automata	Decision Science Incorporated AD-644204	1966
Fogel, L. J., A. J. Owens, and M. J. Walsh	Artificial Intelligence Through Simulated Evolution	New York: Wiley	1966
	On the Evolution of Artificial Intelligence	Proceedings of the Fifth National Symposium on Human Factors, Institute of Electrical and Electronics Engineers, 63-76	1964 May 5-6
	Intelligent Decision-Making Through a Simulation of Evolution	Institute of Electrical and Electronics Engineers Transactions on Human Factors in Electronics, HFE-6, No. 1, 13-23	1965 September
Fralick, S. C.	Learning to Recognize Patterns Without a Teacher	Institute of Electrical and Electronics Engineers Transactions, Information Theory, IT-3, 57-64	1967 January
Friedberg, R., B. Dunham, and J. North	A Learning Machine	IBM Journal of Research and Development, 2-13	1958 January
Frijda, N. H.	Problems of Computer Simulation	Behavioral Science, 12, 59-67	1967 January
Gabor, D.	A New Microscopic Principle	Nature, 161, No. 4098, 777-778	1948 May 15
Gagan, R., F. Garside, L. Metrick, and A. Shortall	Development of a Color Display Capability	Concord, Mass.: Wolf Research and Development Corporation	1968 October

Gold, E. M.	Language Identification in the Limit	Information and Control, 10, 447-474	1967 May
Good, I. J.	Speculations Concerning the First Ultra-Intelligent Machine	Advances in Computers, 6, 31-38	1965
Goto, E.	Difficulties in Realizing Artificial Intelligence	Electronics and Communications in Japan, 46, No. 11, 56-63	1963 November
Gould, I. H.	Some Limitations of Computer-Aided Design	Computer Bulletin, 10, No. 3, 64-68	1966 December
Gray, J. C.	Compound Data Structures for Computer-Aided Design: A Survey	Proceedings of the 22nd Association for Computing Machinery National Conference, 355-365	1967
Green, B. F., C. Chomsky, A. K. Wolf, and K. Laughery	Baseball: An Automatic Question Answerer	Computers and Thought, E. A. Feigenbaum and J. Feldman (editors). New York: McGraw-Hill, 207-216	1963
Greenblatt, R. D., D. E. Eastlake, and S. D. Crocker	The Greenblatt Chess Program	American Federation of Information Processing Proceedings, Fall Joint Computer Conference, 31, 801-810	1967
Guzman, A.	Some Aspects of Pattern Recognition by Computer	Cambridge, Mass.: M.I.T., AD-656041	1967 February
Hall, E. T.	Seeing and Believing	Architectural Review, 149, No. 858	1968 August
	The Silent Language	New York: Doubleday	1959
Hall, P. D.	The Computer and Society	Computer Bulletin, 11, No. 3, 216-219	1967 December
Harper, G. N.	BOP — An Approach to Building Optimized	Proceedings, Association of Computing Machinery 23rd National Conference, 575-583	1968 November
Harper, G. N. (editor)	Computer Applications in Architecture and Engineering	New York: McGraw-Hill	1968

Harrington, J. J.	Operation Research — Relatively New Approach to Managing Man's Environment	New England Journal of Medicine, 275, No. 24, 1342-1350	1966 December 15
Harris, B.	The Limits of Science and Humanism in Planning	American Institute of Planners Journal, 33, No. 5, 324-335	1967a September
	How to Succeed with Computers without Really Trying	American Institute of Planners Journal, 33, No. 1, 11-18	1967b January
	The Uses of Theory in the Simulation of Urban Phenomena	Presented at Highway Research Board meetings, Washington, D.C.	1966 January
	Organizing the Use of Models in Metropolitan Planning	Seminar on Metropolitan Land Use Models, Berkeley, Calif.	1965 March
Hendren, P.	Simulating Architectural Forms	Computer Graphics in Architecture and Design, M. Milne (editor). New Haven: Yale School of Art and Architecture, 38-45	1969
	Experiments in Form, Using Computer Graphics	Santa Barbara: University of California	1968
	A System for Dynamic Simulation, Using Computer Graphics	Stillwater: Oklahoma State University	1967
Hermann, H., and J. C. Kotelly	An Approach to Formal Psychiatry	Perspectives in Biology and Medicine	1967 Winter
Herzberg, J. G.	Architects Open Computer Dialogue	New York Times, VIII, 1:7	1968 April 21
Herzog, B.	Computer Graphics for Designers	Emerging Concepts in Computer Graphics, D. Secrest and J. Nievergelt (editors). New York: W. A. Benjamin, 189-230	1968
Hodes, I.	Machine Processing of Line Drawings	Lexington, Mass.: M.I.T. Lincoln Laboratory, Group Report 54G0028	1961 March

Hogan, D. L.	Speech as Computer Input and Output	1966 Institute of Electrical and Electronics Engineers International Convention Record, Part 3, 91-93	1966
Holmes, J. N.	Machines that Talk	Science Journal, 75-80	1968 October
Hook, S. (editor)	Dimensions of Mind Proceedings	New York: New York University Press	1960
Horgan, T. B.	Computer Graphics: An Overview	Reprographics, V, 6	1967 June
Hormann, A.	Shimoku Posed as an Assignment Task	Systems Development Corporation, AD-643264	1966 September 28
	Designing a Machine Partner — Prospects and Problems	IBM Corporation, AD-626173	1965 October 15
	Gaku, An Artificial Student	Behavioral Science, 10, 88-107	1965 January
	How a Computer System Can Learn	Institute of Electrical and Electronics Engineers Spectrum, 1, No. 7, 110-119	1964 July
	Programs for Machine Learning, Part II	Information and Control, 7, 55-57	1964 March
	Programs for Machine Learning, Part I	Information and Control, 5, 347-367	1962 December
Hosaka, M.	A Theory and Design of Free-Formed Surface	Information Processing in Japan, 7, 54-61	1967
Hosken, F. P.	Our Cities of Tomorrow	The Boston Herald, Sunday Magazine Section, 7	1968 January 7
Huberman, B. J.	A Program to Play Chess End Games	California: Stanford University, AD-673971	1968 August 19
Hulten, K. G. P.	The Machine	New York: The Museum of Modern Art	1968
Huxley, A.	Brave New World	New York: Harper & Row	1939
Imbert, J., and B. Lagune	Spécimen de Dessin Automatique de Contours d'Ilots et d'une Typologie	Aix-en-Provence, France: Centre d'Etudes Techniques de l'Equipment	1968

Johnson, A. R.	The Self-Organizing Interface	Institute of Electrical and Electronics Engineers, Group on Man Machine Systems	1969a September 8-12
	Self-Organizing Control in Prosthetics	To be presented at the Third International Symposium on External Control of Human Extremities	1969b August 25-30
	Organization, Perception, and Control in Living Systems	Industrial Management Review, 10, No. 2, 1-16	1969c Winter
	A Structural, Preconscious Piaget: Heed Without Habit	Proceedings, National Electronics Conference, 23	1967
Johnson, C. I.	Principles of Interactive Systems	IBM Systems Journal, 7, Nos. 3 and 4, 147-174	1968
Johnson, D. L., and A. L. Kobler	Man-Computer Relationships	Science, 139, 1231-1232	1963 March 22
Johnson, T. E.	Space Arrangement	Architectural Design, 39, 509	1969 September
	A Mass Storage Relational Data Structure for Computer Graphics and other Arbitrary Data Storage	Cambridge, Mass.: Department of Architecture, M.I.T.	1967
	Sketchpad III: A Computer Program for Drawing in Three Dimensions	American Federation of Information Processing Proceedings, Spring Joint Computer Conference, 23, 347-353	1963
Juenger, F. G.	The Failure of Technology; Perfection Without Purpose	Hinsdale, Ill.: Regnery	1949
Julesz, B.	Toward the Automation of Binocular Depth Perception	Proceedings of the International Federation of Information Processing, 9, No. 3, 439-444	1962
Kamnitzer, P.	Computer Aid to Design	Architectural Design, 39, 507-508	1969 September
Keast, D. N.	Survey of Graphic Input Devices	Machine Design, 39, No. 18, 114-120	1967 August 3

Kellog, C. H.	CONVERSE—A System for On-Line Description and Retrieval of Structural Data Using Natural Language	Systems Development Corporation, SP-2635	1967 May 26
Ketchpel, R. D.	Direct-View Three-Dimensional Display Tube	Institute of Electrical and Electronics Engineers Transactions of the Professional Technical Group on Electron Devices, ED-10, No. 5, 324-328	1963 September
Kirsch, R. A.	Experiments in Processing Pictorial Information with a Digital Computer	National Building Studies Report, No. 5713	1957 December 15
Klopf, A. K.	Evolutionary Pattern Recognition Systems	Illinois University Research Report, AD-637492	1965 November 1
Knowlton, K. C.	Computer-Animated Movies	Emerging Concepts in Computer Graphics, D. Secrest and J. Nievergelt (editors). New York: W. A. Benjamin, 343-370	1968
Kochen, M.	Automatic Question-Answering of English-like Questions about Simple Diagrams	Radio Corporation of America Laboratories, AD-670545	1968 May
	Group Behavior of Robots	Computers and Automation, 6, No. 3, 16-21, 48	1957 March
Kock, W. E.	Hologram Television	Institute of Electrical and Electronics Engineers, Proceedings, 54, No. 2, 331	1966 February
Koffman, E. B.	Learning Games through Pattern Recognition	Institute of Electrical and Electronics Engineers Transactions, Systems Science and Cybernetics, SSC-4, 12-16	1968 March
Kolers, P. A., and M. Eden (editors)	Recognizing Patterns; Studies in Living and Automatic Systems	Cambridge, Mass.: The M.I.T. Press	1968
Krulee, G., D. Kuck, D. Landi, and D. Manelski	Natural Language Inputs for a Problem-Solving System	Behavioral Science, 9, 281-288	1964 July 9

Laning, J. H., Jr., and R. H. Battin	Random Processes in Automatic Control	New York: McGraw-Hill	1956
Leith, E. N., A. Kozma, and N. Massey	Hologram Visual Displays	Journal of the Society of Motion Picture and Television Engineers, 75, 323	1966
Leith, E. N., and J. Upatnieks	Holograms, Their Properties and Uses	Journal of the Society for Photographic Instrumentation Engineers, 4, 3-6	1965
Lesem, L. B., P. M. Hirsch, and J. A. Jordan, Jr.	Computer Synthesis of Holograms for 3-D Display	Houston: IBM Houston Scientific Center Report 320-2327	1968 January
Lettvin, J. Y., H. Maturana, W. S. McCulloch, and W. Pitts	What the Frog's Eye Tells the Frog's Brain	Proceedings of the Institute of Radio Engineers, 47, 1940-1951	1959
Lewis, A. O., Jr. (editor)	Of Men and Machines	New York: Dutton	1963
Licklider, J. C. R.	Libraries of the Future	Cambridge, Mass.: The M.I.T. Press	1965a
	Man-Computer Communication—Introduction	Communication Processes, F. A. Geldard (editor). New York: Pergamon	1965b
	Problems in Man-Computer Communications	Communication Processes, F. A. Geldard (editor). New York: Pergamon	1965c
	Man-Computer Symbiosis	Institute of Radio Engineers Transactions on Human Factors in Electronics, HFE-1, No. 1, 4-10	1960 March
Lindgren, N.	Art and Technology, Steps Toward a New Synergism	Institute of Electrical and Electronics Engineers Spectrum, 6, No. 4, 59-68	1969 April
	Directions for Speech Research	Institute of Electrical and Electronics Engineers Spectrum, 5, No. 3, 83-92	1968 March
	Human Factors in Engineering—Part II	Institute of Electrical and Electronics Engineers Spectrum, 3, No. 4., 62-72	1966a April
	Human Factors in Engineering—Part I	Institute of Electrical and Electronics Engineers Spectrum, 3, No. 3, 132-139	1966b March

Lindgren, N.	Machine Recognition of Human Language—Part II	Institute of Electrical and Electronics Engineers Spectrum, 2, No. 4, 45-59	1965a April
	Machine Recognition of Human Language—Part I	Institute of Electrical and Electronics Engineers Spectrum, 2, No. 3, 114-136	1965b March
Lindheim, R.	Computers and Architecture	Landscape, 14, No. 3, 8-11	1965 Spring
Lipner, S. B.	Requirements for the Development of Computer-Based Urban Information Systems	American Federation of Information Processing Proceedings,Spring Joint Computer Conference, 34, 523-528	1969
Loehlin, J. C.	Machines with Personality	Science Journal, 4, No. 10, 97-101	1968 October
Logcher, R. D., and G. M. Sturman	The Structural Design Language; a Design System for a Process Approach to Design	Cambridge, Mass.: School of Engineering, M.I.T.	1966
Loginou, V. N.	The Application of Electronic Computers to Planning	Mekhaniz. I Avtomatiz, ProvIA-VA9, 32-35	1963
Lovelace, A. A.	Translators Notes to an Article on Babbage's Analytical Engine	Scientific Memoirs, 3, 691-731	1842
Ludwig, M. E.	Architecture in the McLuhan Age	American Institute of Architects Journal,48,No.1, 37-38	1967 August
	Prejudice and the Computer	American Institute of Architects Journal, 46, No. 1, 70	1966 July
McCarthy, J.	Information	Scientific American, 215, No. 3, 65-95	1966 September
	Programs with Common Sense	Mechanization of Thought Processes, 1. London: Her Majest's Stationery Office	1959
McCarthy, J., E. Feigenbaum, and A. Samuel	Project Technical Report	California: Stanford University, AD-677528	1968 September 13
McCulloch, W. S.	Lekton	Communication: Theory and Research, L. Thayer (editor). Springfield, Ill.: Charles C. Thomas	1967

McCulloch, W. S.	Embodiments of Mind	Cambridge, Mass.: The M.I.T. Press	1965
	Toward Some Circuitry of Ethical Robots or an Observational Science of the Genesis of Social Evaluation in the Mind-Like Behavior of Artifacts	Acta Biotheoretica, 11, 147-156	1956
McCulloch, W. S., and W. M. Brodey	The Biological Sciences	The Great Ideas Today. Chicago: Encyclopedia Britannica	1966
McCulloch, W. S., and W. Pitts	Machines that Know and Want	Brain Behavior: A Symposium,W. C. Halstead (editor), Comparative Psychology Monographs, 20, No. 1. Berkeley: University of California Press	1950
	How We Know Universals	The Perception of Auditory and Visual Forms, Bulletin of Mathematical Biophysics. 9, 127-147	1947 June
	A Logical Calculus of the Ideas Immanent in Nervous Activity	Bulletin of Mathematical Biophysics, 5, No. 4, 115-134	1943
MacKay, D. M.	Mind-Like Behavior in Artifacts	British Journal of Philosophy of Science, 2, 105-121	1951
McLuhan, M.	Understanding Media: The Extensions of Man	New York: McGraw-Hill	1965
Maguire, H. T., and W. Arnold	Intelligent Robots: Slow Learners	Electronics, 40, No. 9, 117-120	1967 May 1
Makowski, Z. S.	Space, Structure and Electronic Computers	Architectural Design, 36, No. 1, 8-9	1966 January
Manheim, M. L.	Problem-Solving Processes in Planning and Design	Design Quarterly 66/67, 31-40	1966 December
	Role of the Computer in the Design Process	Building Research, 3, No. 2, 13-17	1966 March-April
Mann, R. W., and S. A. Coons	Computer-Aided Design	McGraw-Hill Yearbook of Science and Technology. New York: McGraw-Hill, 1-9	1965

Meeker, R. J., and G. H. Shure	Updating some Ground Rules for Man-Machine Simulation	Systems Development Corporation, AD-672783	1968 April 25
Meier, R. L.	The Metropolis as a Transactions-Maximizing System	Daedalus, 97, No. 4, 1292-1314	1968 Fall
Meier, R. L., and R. D. Duke	Gaming Simulation for Urban Planning	American Institute of Planners Journal, 32, No. 1, 3-17	1966 January
Melick, L. F.	The Computer Talks Back	Data Processing Magazine, 8, No. 10, 58-62	1966 October
Mezie, L.	Electronic Computer: A New Tool for the Artist	Arts Canada, 24, 20-21	1967 February
	Conversation with a Computer	Datamation, 13, No. 1, 57-58	1967 January
Michael, D.	Letter	Science, 139, 1231	1963 March
Michie, D.	Machines that Play and Plan	Science Journal, 4, No. 10, 83-88	1968 October
Miller, A. R.	The National Data Center and Personal Privacy	The Atlantic Monthly, 220, No. 5, 53	1967 November
Miller, G. A.	Man-Computer Interaction	Communication Processes, F. A. Geldard (editor), New York: Pergamon	1965
Miller, R. B.	Psychology for a Man-Machine Problem-Solving System	International Business Machine Corporation, AD-640283	1965 February 19
Miller, R. I.	Invasion of Privacy by Computer	Lex et Scientia, 5, No. 1, 18-24	1968 February
Milne, A.	Waiting for the Takeover	Spectator, 210, 649	1963 May 17
Milne, M. (editor)	Computer Graphics in Architecture and Design	New Haven: Yale School of Art and Architecture,	1969
Minsky, M.	I Think, Therefore I Am	Psychology Today, 2, No. 11, 30-32	1969 April
	Machines Are More Than They Seem	Science Journal, 4, No. 10, 3-43	1968 October
	Computation: Finite and Infinite Machines	Englewood, N. J.: Prentice-Hall	1967

Minsky, M.	Artificial Intelligence	Scientific American, 215, No. 3, 246-260	1966 September
	Matter, Mind, and Models	Proceedings, International Federation of Information Processing Congress, I, 45-50	1965
	Steps Towards Artificial Intelligence	Institute of Radio Engineers, Proceedings, 49, No. 1, 8-30	1961 January
Minsky, M. (editor)	Semantic Information Processing	Cambridge, Mass.: The M.I.T. Press	1968
Minsky, M., and S. Papert	Perceptrons	Cambridge, Mass.: The M.I.T. Press	1969
Minsky, M., and O. G. Selfridge	Learning in Neural Nets	Proceedings of the Fourth London Symposium on Information Theory, C. Cherry (editor). New York: Academic Press	1961
Moore, G. (editor)	Emerging Methods in Environmental Design and Planning	Cambridge, Mass.: The M.I.T. Press	1970 forthcoming
Moullen, F. J.	Chess and the Computer	Datamation, 14, 65-68	1968 April
Mumford, E., T. B. Ward, and B. T. Bateford	Computers: Planning for People	Manchester, England: Manchester Business School	1968
Mumford, L.	The Myth of the Machine	NewYork: Harcourt, Brace & World	1967
	Utopia, The City and The Machine	Daedalus, 94, No. 2, 271-292	1965 Spring
	Techniques and Civilization	New York: Harcourt, Brace & World	1934
Muses, C. A. (editor)	Aspects of the Theory of Artificial Intelligence	New York: Plenum Press	1962
Myer, J., and R. I. Krauss	Design: A Case History	Cambridge, Mass.: M.I.T. Center for Building Research	1967
Myer, T. H., and R. I. Krauss	Architectural Practice and the Computer Revolution	Architectural Design, 37, No. 4, 193-195	1967 April

Naylor, T. H., J. L. Balintfy, and D. S. Burdick	Computer Simulation Techniques	New York: Wiley	1966
Negroponte, N. P.	URBAN5—A Machine that Discusses Urban Design	Emerging Methods in Environmental Design and Planning, G. Moore (editor). Cambridge, Mass.: The M.I.T. Press	1970 (forthcoming)
	URBAN5: An Experimental Urban Design Partner	Computer Graphics in Architecture and Design, M. Milne (editor). New Haven: Yale School of Art and Architecture	1969a
	Architecture Machine	Architectural Design, 39, 510	1969b September
	Towards a Humanism Through Machines	Architectural Design, 39, 511-512, reprinted from Technology Review, 71, No. 6, April 1969, Copyright 1969	1969c September
	Towards a Humanism Through Machines	Technology Review, 71, No. 6, 2-11	1969d April
	Humanism Through Machines	The Canadian Architect, 14, No. 4, 29-34	1969e April
	Toward a Theory of Architecture Machines	American Institute of Architects Journal, 51, No. 3, 71-74	1969f March
	The Computer Simulation of Perception During Motion in the Urban Environment	Cambridge, Mass.: Master's Thesis in Architecture, M.I.T.	1966
Negroponte, N. P., and L. B. Groisser	Environmental Humanism Through Robots	Proceedings of the First Annual Environmental Design Research Association Conference, H. Sanoff, and S. Cohn (editors). Raleigh: Design Research Laboratory, North Carolina State University	1970 (forthcoming)
	URBAN5	Ekistics, 24, No. 142, 289-291	1967a September

Negroponte, N. P. and L. B. Groisser	URBAN5: An On-Line Urban Design Partner	IBM Report, 320-2012. Cambridge, Mass.	1967b June
Newell, A., and G. Ernst	The Search for Generality	Proceedings International Federation of Information Processes, I., W. Kalenik (editor). Washington, D.C.: Spartan Books	1965
Newell, A., J. C. Shaw, and H. A. Simon	Chess-Playing Programs and the Problem of Complexity	Computers and Thought, E. A. Feigenbaum and J. Feldman (editors). New York: McGraw-Hill, 39-70	1963
	A General Problem Solving Program for a Computer	Computers and Automation, 8, No. 7, 10-17	1959 July
	A Variety of Intelligent Learning in a General Problem Solver	The Rand Corporation Report No. P-1742	1959 July 6
Newell, A., and H. A. Simon	Overview: Memory and Process in Concept Formation	Concepts and the Structure of Memory, B. Kleinmuntz (editor). New York: Wiley, 241-262	1967
	Computer Simulation of Human Thinking	Science, 134, 2011-2017	1961 December
	The Logic Theory Machine	Institute of Radio Engineers Transactions on Information Theory, IT-2, No. 3, 61-79	1956 September
Newman, C., and L. Uhr	BOGART: A Discovery and Induction Program for Games	Proceedings of the 20th Association for Computing Machinery National Conference, 176-186	1965
Newman, W. M.	An Experimental Program for Architectural Design	Computer Journal, 9, No. 1, 21-26	1966 May
Noll, A. M.	The Digital Computer as a Creative Medium	Institute of Electrical and Electronics Engineers Spectrum, 4, No. 10, 89-95	1967 October
Orr, W. D.(editor)	Conversational Computers	New York: Wiley	1968
Overton, R. K.	Intelligent Machines and Hazy Questions	Computers and Automation, 14, No. 7, 26-30	1965 July

Papert, S.	Some Mathematical Models of Learning	Proceedings of the Fourth London Symposium on Information Theory, C. Cherry (editor). New York: Academic Press	1961
Papert, S., and R. McNaughton	On Topological Events	Theory of Automata, Ann Arbor: University of Michigan Summer Conferences	1966
Pask, G.	The Architectural Relevance of Cybernetics	Architectural Design, 39, 494-496	1969 September
	Cybernetics and Education	System Research Ltd., AD-657806	1967 August
	A Discussion of Artificial Intelligence and Self-Organization	Advances in Computers, 5, 110-218	1964
	The Conception of a Shape and the Evolution of a Design	Conference on Design Methods, J. C. Jones and D. G. Thornley (editors). Oxford: Pergamon Press, 153-168	1963
	The Simulation of Learning and Decision-Making Behavior	Aspects of the Theory of Artificial Intelligence, C. A. Muses (editor). New York: Plenum Press, 165-210	1962
	My Prediction for 1984	Prospect. London: Hutchinson	1962
Passonneau, J., and R. Wurman	Urban Atlas: 20 American Cities	Cambridge, Mass.: The M.I.T. Press	1966
Pierce, J. R.	Communication	Daedalus, 96, No. 3, 909-921	1967 Summer
	Communications, Technology and the Future	Daedalus, 94, No. 2, 506-517	1965 Spring
Pool, I. d.S.	Simulating Social Systems	International Science and Technology, No. 27, 62-71	1964 March
Porter, W., K. Lloyd, W. McMains, and A. Fleisher	DISCOURSE	Architectural Design, 39, 510	1969 September
Prince, M. D.	Man-Computer Graphics for Computer-Aided Design	Institute of Electrical and Electronics Engineers, Proceedings, 54,No.12, 1698-1708	1966 December

Quilan, M. R.	Semantic Memory	Bolt, Beranek, and Newman Incorporated, AD-641671	1966 October
Quinlan, J. R.	A Task-Independent Experience-Gathering Scheme for a Problem Solver	Proceedings of the International Joint Conference on Artificial Intelligence, 193-198	1969 May 7-9
	A FORTRAN IV General-Purpose Deductive Program	Los Angeles: University of California, AD-641194	1966 August
Quinlan, J. R., and E. B. Hunt	A Formal Deductive Problem-Solving System	Journal of the Association of Computing Machinery, 15, 625-646	1968 October
Rabiner, L. R.	Speech Synthesis by Rule: An Acoustic Domain Approach	Bell System Technical Journal, 47, No. 1, 17-37	1968
Rapoport, A.	Fights, Games, and Debates	Ann Arbor: University of Michigan Press	1960
Rapoport, A., and R. Kantor	Complexity and Ambiguity in Environmental Design	Journal of American Institute of Planners, 33, No. 3, 210-221	1967 July
Raphael, B.	Research on Intelligent Question-Answering Systems	California: Stanford Research Institute, AD-671970	1968 May
	A Computer Program which Understands	American Federation of Information Processing Proceedings, Fall Joint Computer Conference, 577-590	1964
Raser, J. R.	Simulation and Society: An Exploration of Scientific Gaming	Boston: Allyn and Bacon, Inc.	1969
Reddy, D. R., and P. J. Vicens	A Computer with Hands, Eyes, and Ears	American Federation of Information Processing Proceedings, Fall Joint Computer Conference, 33, 329-338	1968
Rice, E. L.	The Adding Machine	London: French	1929
Rittel, H.	The Universe of Design	Berkeley: Institute of Urban and Regional Development, University of California	1964

Robequain, D.	Informatique, Action Pilote	Banque de Données Urbaines. Aix-en-Provence, France: Centre d'Etudes Techniques de l'Equipment	1968
Roberts, L. G.	The Lincoln Wand	American Federation of Information Processing Proceedings, Fall Joint Computer Conference, 29, 223-227	1966
	Machine Perception of Three Dimensional Solids	Cambridge, Mass.: Ph.D. Thesis, M.I.T.	1963 June
Rosen, C. A.	Machines that Act Intelligently	Science Journal, 4, No. 10, 109-114	1968 October
Rosenblueth, A., N. Wiener, and J. Bigelow	Behavior, Purpose, and Teleology	Philosophy of Science, 10, 18-24	1943
Rosenfeld, A.	Picture Processing by Computer	Maryland: Maryland University, AD-672775	1968 June
Ross, D. T.	The AED Approach to Generalized Computer-Aided Design	Proceedings of the 23rd Association for Computing Machinery National Conference, 367-385	1967
Ross, D. T., and S. A. Coons	Investigations in Computer-Aided Design	Cambridge, Mass.: M.I.T. Report 8436-IR-Ad-26573	1961 November
Roy, A. E.	On a Method of Storing Information	Bulletin of Mathematical Biophysics, 22, 139-168	1960
Rudofsky, B.	Architecture without Architects	New York: Museum of Modern Art	1964
Runge, A.	In Search of Urban Expertise	Progressive Architecture, No. 9, 124-129	1969 September
Samuel, A. L.	Some Studies in Machine Learning Using the Game of Checkers, Part II	IBM Journal of Research and Development, 11, No. 4, 601-618	1967
	Some Moral and Technical Consequences of Automation—A Refutation	Automation: Implications for the Future, M. Philipson (editor). New York: Random House, Vintage Books, 174-179	1962

Samuel, A. L.	Programming Computers to Play Games	Advances in Computers, 1, 165-192	1960
	Some Studies in Machine Learning Using the Game of Checkers, Part I	IBM Journal of Research and Development, 3, No. 3, 210-223	1959
Sanoff, H.	A Systems Approach to Design	Raleigh: Design Research Laboratory, North Carolina State University	1968
	Techniques of Evaluation for Designers	Raleigh: School of Design, North Carolina State University	1968
Sanoff, H., and S. Cohn (editors)	Proceedings of the First Annual Environmental Design Research Association Conference	Raleigh: Design Research Laboratory, North Carolina State University	1970 (forthcoming)
Sayre, K. M.	Recognition: A Study in the Philosophy of Artificial Intelligence	Notre Dame, Ind.: University of Notre Dame	1965
Scriven, M.	The Complete Robot	Dimensions of Mind Proceedings, S. Hook (editor). New York: New York University Press	1960
Secrest, D., and J. Nievergelt (editors)	Emerging Concepts in Computer Graphics	New York: W. A. Benjamin	1968
Selfridge, O. G., and J. R. Kelly	Sophistication in Computers: A Disagreement	Institute of Radio Engineers, Transactions on Information Theory, IT-8, No. 2, 78-80	1962 February
Selfridge, O. G., and U. Neisser	Pattern Recognition by Machine	Computers and Thoughts, E. A. Feigenbaum and J. Feldman (editors). New York: McGraw-Hill, 237-250	1963
	Pattern Recognition by Machine	Scientific American, 1203, No. 3, 60-68	1960 August
Senders, J. W.	Information Storage Requirements for the Contents of the World's Libraries	Science, 141, 1067-1068	1963 September
Servan-Schreiber, J.-J.	Le Défi Americain	Paris: Donoil	1967

Shannon, C. E.	A Chess Playing Machine	The World of Mathematics, 4, J. R. Newman (editor). New York: Simon and Schuster	1956
Shannon, C. E., and E. Weaver	Computers and Automata	Institute of Radio Engineers, Proceedings, 41, No. 10, 1234-1241	1953 October
Shubik, M.	Information, Rationality, and Free Choice in a Future Democratic Society	Daedalus, 96, No. 3, 771-778	1967 September
Siders, R. A.	Computer-Aided Design	Institute of Electrical and Electronics Engineers Spectrum, 4, No. 11, 84-92	1967 November
Silverstone, S. M., and C. W. Rusch	The Medium is Not the Solution	American Institute of Architects Journal, 48, No. 6, 61-67	1967 December
Simon, H. A.	The Sciences of the Artificial	Cambridge, Mass.: The M.I.T. Press	1969
	Research for Choice	Environment and Policy, W. R. Ewald, Jr. (editor). Bloomington, Ind.: Indiana University Press	1968
	Modeling Human Mental Processes	Proceedings of the Western Joint Computer Conference, 111-120	1961
Simon, H. A., and G. W. Baylor	A Chess Mating Combinations Program	American Federation of Information Processing Proceedings, Spring Joint Computer Conference, 28, 431-447	1966
Skinner, B. F.	Science of Human Behavior	New York: Macmillan	1953
Skinner, F.	History of Computer Graphics	Computer Graphics in Architecture and Design, M. Milne (editor). New Haven: Yale School of Art and Architecture	1969
	Computer Graphics—Where Are We?	Datamation, 12, No. 5, 28-31	1966 May
Snediker, J. M.	A Self-Organizing Program for Describing Concepts	Proceedings of the 20th Association for Computing Machinery National Conference, 101-118	1965

Solomonoff, R. J.	The Search for Artificial Intelligence	Electronics and Power, 14, 8-11	1968 January
	Some Recent Work in Artificial Intelligence	Institute of Electrical and Electronics Engineers Proceedings, 54, No. 12, 1687-1697	1966 December
	An Inductive Inference Machine	Institute of Radio Engineers National Convention Record, Part II, 56-62	1957
Spencer, D. D.	Game Playing with Computers	Computers and Automation, 17, 38-40, 42	1968 August
Spring, B. P.	Will Computers Design Our Buildings?	Architectural Forum, 117, 122-125	1962 October
Stea, D.	Mediating Medium	American Institute of Architects Journal, 48, No. 6, 67-70	1967 December
	Space, Territory and Human Movements	Landscape, 13-16	1965 Autumn
Steel, T. B., Jr.	Artificial Intelligence Research—Retrospect and Prospect	Computers and Automation, 16, No. 1, 22-24	1967 January
Steinitz, C., and P. Rogers	A Systems Analysis Model of Urbanization and Change: An Experiment in Interdisciplinary Education	Cambridge, Mass.: Graduate School of Design, Harvard University	1968
Stockham, T. G., Jr.	Some Methods of Graphical Debugging	Proceedings of the IBM Scientific Computing Symposium on Man-Machine Communication, 57-72	1965
Storm, H. O.	Eolithism and Design	Colorado Quarterly, 1, No. 3	1953 Winter
Stroke, G. W., and R. C. Zech	White-Light Reconstruction of Color Images from Black and White Volume Holograms Recorded on Sheet Film	Applied Physics Letters, 9, No. 5, 215-217	1966 September 1
Studer, R. G.	The Dynamics of Behavior-Contingent Physical Systems	Proceedings of the Portsmouth Symposium on Design, Portsmouth, England	1967 December

Studer, R. G.	Architectural Programming, Environmental Design, and Human Behavior	Journal of Social Issues, 22, 127-136	1966
	On Environmental Programming	Architectural Association Journal, 81, No. 902, 290-296	1966 May
Sutherland, I. E.	A Head Mounted Three Dimensional Display	American Federation of Information Processing Proceedings, Fall Joint Computer Conference, 33, 757-766	1968
	Computer Graphics: Ten Unsolved Problems	Datamation, 12, No. 5, 22-27	1966 May
Sutherland, N. S.	Machines like Men	Science Journal, 4, No. 10, 44-49	1968 October
Sutro, L. L.	Assembly of Computers to Command and Control a Robot	American Federation of Information Processing Proceedings, Spring Joint Computer Conference, 34, 113-138	1969
Taube, M.	Computers and Common Sense	New York: McGraw-Hill	1961
	Computers and Game-Playing	Science, 132, 555-557	1960 August 26
Taylor, W. K.	Machines that Learn	Science Journal, 4, No. 10, 102-108	1968 October
Teague, L. C., Jr.	Research in Computer Applications to Architecture	Computer Applications in Architecture and Engineering, G. N. Harper (editor). New York: McGraw-Hill, 191-216	1968
Teicholz, E. D.	Architecture and the Computer	The Architectural Forum, 129, No. 2, 58-62	1968 September
Thompson, D. W.	On Growth and Form	Cambridge, England: Cambridge University Press	1917
Thornhill, D. E.	An Integrated Hardware-Software System for Computer Graphics in Time-Sharing	ESL-R-356. Cambridge, Mass.: M.I.T., Electronic Systems Laboratory	1968

Throop, T. A.	Learning and Artificial Intelligence Accomplished by Computer Programs	Computers and Automation, 15, No. 11, 28-33	1966 November
Tonge, F. M.	A View of Artificial Intelligence	Proceedings of the 21st Association for Computing Machinery National Conference, 379-382	1966
Turing, A. M.	Computing Machinery and Intelligence	Mind, 59, No. 236, 433-460	1950 October
Uhr, L., and M. Kochen	MIKROKOSMS and Robots	Proceedings of the International Joint Conference on Artificial Intelligence, 541-556	1969 May 7-9
Uttley, A. M.	Conditional Probability Machines and Conditioned Reflexes	Automata Studies, C. E. Shannon and J. McCarthy (editors). Princeton: Princeton University Press, 253-276	1956
Vachon, D. A.	Closing the Intuition Gap	Architectural and Engineering News, 11, No. 5, 29-31	1969 May
Venturi, R.	Complexity and Contradiction in Architecture	New York: Museum of Modern Art	1966
Von Foerster, J. (editor)	Cybernetics, Circular Causal, and Feedback Mechanisms in Biological and Social Systems	New York: Josiah Macy, Jr. Foundation	1949
Von Neumann, J.	The General and Logical Theory of Automata	The World of Mathematics, 4, J. R. Newman (editor). New York: Simon and Schuster	1956
Warntz, W.	Computer Graphics and Spatial Analysis	Cambridge, Mass.: Graduate School of Design, Harvard University	1969
Weizenbaum, J.	Contextural Understanding by Computers	Communications of the Association for Computing Machinery, 10, No. 8, 474-480	1967 August
	ELIZA—A Computer Program for the Study of Natural Language Communication Between Man and Machine	Communications of the Association for Computing Machinery, 9, No. 1, 36-45	1966 January

Westervelt, F., and D. Smith	Touch-Tone Telephone Users' Guide	Ann Arbor: University of Michigan, Computing Center Memo #33	1968
Wheeler, C. H., Jr.	Today's New Tools for Tomorrow's Practice	Architectural Record, 142, No. 12, 93-94	1967 December
Whorf, B. L.	Language, Thought, and Reality	J. E. Carroll (editor). Cambridge, Mass.: The M.I.T. Press	1956
Wiener, N.	God and Golem, Inc.	Cambridge, Mass.: The M.I.T. Press	1964
	Some Moral and Technical Consequences of Automation	Automation: Implications for the Future, M. Philipson (editor). New York: Random House, Vintage Books, 162-173	1962
	Some Moral and Technical Consequences of Automation	Science, 131, 1355-1358	1960 May 6
	The Human Use of Human Beings	Cybernetics and Society. Boston: Houghton Mifflin	1950
	Cybernetics	Cambridge, Mass., and New York: The M.I.T. Press and Wiley	1948
Wolin, J. L.	The Development of Computer-Aided Heuristic Game Models in City and Regional Planning	A paper prepared for Professor Raymond DiPasquale in Computer Application in Architecture at Cornell University, Ithaca, New York	1968
Woodruff, R.	Industrial Applications of Computer Graphics	Computer Graphics in Architecture and Design, M. Milne (editor). New Haven: Yale School of Art and Architecture	1969
Yershov, A. P.	One View of Man-Machine Interaction	Journal of the Association for Computing Machinery, 12, No. 3, 315-325	1965 July
Yovits, M. C., G. T. Jacobi, and G. D. Goldstein (editors)	Self-Organizing Systems	Washington, D.C.: Spartan Books	1962