ARCAID

The ARChitect's Computer Graphics AID

by

Robert Wehrli

Max J. Smith

Edward F. Smith

June 1970 UTEC-CSc-70-102

This research was supported in part by the University of Utah Computer Science Division and the Advanced Research Projects Agency of the Department of Defense, monitored by Rome Air Development Center, Griffiss Air Force Base, New York 13440, under contract AF(602)-4277. ARPA Order No. 829.

TABLE OF CONTENTS

Abstract v

Preface 1

I. ARCAID - The ARChitect's Computer Graphics AID 3

1.1 The On-line Console 5
1.2 The Interfaced Computer 6
1.3 The Project File 7
1.4 The Plotter 7
1.5 Numeric Control Machine 8
1.6 General Purpose Processes 9
1.7 Uses of the General Purpose Processes 10
1.8 Object System Processes 10
1.9 Communication among Object System Processes 10
1.10 Communication between the Object System Processes and the Project File 12
1.11 Attribute System Processes 13
1.12 Communication among Attribute System Processes 13
1.13 Communication between the Attribute System Processes and the Project File 14
1.14 The ARCAID Library 15

II. The SPACEFORM Graphics Language 16

2.1 The Rationale for Spaceforms 16
2.2 Topology 22
2.3 Perspectives 22
2.4 Heirarchical Organization 23
2.5 Associative Memory 23

III. Housekeeping Manipulations 25

3.1 Log In 25
3.2 Touch 30
3.3 Place 30
3.4 Execute 31
3.5 Recover 31
3.6 Erase 31
3.7 Delete 32
3.8 Store 32
3.9 Constrain; Constraints 33
3.10 Name 33
3.11 Label 34
3.12 Duplicate 34
3.13 Flop 35
3.14 Log Out 36

TABLE OF CONTENTS (CONTINUED)

IV. Shaping Manipulations 37

4.1 Contract-Expand 37
4.2 Size 39
4.3 Cut 40
4.4 Draw 42
4.5 Grid 42
4.6 Shape a Surface 43
4.7 Retrieve a Surface 44
4.8 Reshape a Surface 45
4.9 Revolve 46

V. Assembling Manipulations 48

5.1 Translate 48
5.2 Assemble at an angle 49
5.3 Make Coordinate 49
5.4 Make Colinear 50
5.5 Make Coplanar 50
5.6 Pack 50
5.7 Manipulate Clusters and Components 50

VI. Viewing Manipulations 53

6.1 Scroll 53
6.2 Enlarge; Reduce 53
6.3 Rotate 54
6.4 Zoom 55
6.5 Pan 57
6.6 Scan 57
6.7 Cut to View 58
6.8 Sectional Zoom 58

VII. Orienting Manipulations 59

7.1 Introduction 59
7.2 Orienting 66

VIII. Miscellaneous Manipulations 68

8.1 Dimension 68
8.2 Plot 69

IX. Elements in the SPACEFORM Language and ARCAID 70

9.1 Element 70
9.2 Letter 70

TABLE OF CONTENTS (CONTINUED)

9.3 Number 70
9.4 Primitives 70
9.4.1 Point, Corner 70
9.4.2 Line, Edge, False Edge, Axis 70
9.4.3 Curve 71
Boundary Curve
Primitive Curve
Drawn Curve
9.4.4 Surface, Plane 71
Surface
Plane
Curved Surface
Boundary Defined Surface
Face
9.4.5 Angle 72
9.4.6 Surface Net 72
Net
Net Node
9.5 Basic Spaceform 72
9.5.1 Basic Spaceforms 73
Cuboid
Cylinder
Sphere
Triangular Rod
Hexagonal Rod
9.5.2 Types 75
Frame
Box
Solid
9.6 Shaped Spaceform 78
9.6.1 Spaceform of Revolution 78
9.7 Object 78

References 80

Appendix A - Building Systems 85

A.1.1 Introduction 85
A.1.2 Building Systems 85
A.1.3 Object Systems 87
A.1.4 Attribute Systems 89
A.2 The Structures System Process 92
A.2.1 Introduction 92
A.2.2 Input for Structures 94
A.3 Analysis Procedures 99
A.4 Output 99

Appendix B - Definitions 101

ARCAID

ABSTRACT

ARCAID—The ARChitect's Computer Graphics AID—is one part of a two-part research program at the University of Utah under the direction of David C. Evans. ARCAID is a specification for the organization of computer processes including data and procedures for the use of architects, engineers, and others in design. As the second part of this research C. Stephen Carr is developing the complex data structure which supports ARCAID. This data structure includes a graphics FORTRAN, a compiler-compiler, an associative memory, and a tree structure for organizing the data for a building scheme.

ARCAID is an interactive computer graphics system relying on the Cathode Ray Tube (CRT) for feedback and a typing keyboard, stylus and tablet, track ball, and zoom pedal for input. The picture at the CRT is refreshed by a small computer, and manipulations of alphanumerics and graphic elements are handled by a large computer. ARCAID envisions the design of a building from first briefing and schematics through construction without the use of paper.

ARCAID incorporates a computer graphics language called SPACEFORM. It provides for graphic elements and procedures by which elements are manipulated. The basic graphic elements, "spaceforms," are built up from such primitives as points, lines, and surfaces. Basic spaceforms may be shaped and assembled for more complex (shaped) spaceforms and objects. Proposed manipulations include housekeeping, shaping, assembling, viewing, orienting, and miscellaneous manipulations. Spaceforms are topologically

described with respect to geometry and attributes. By geometric topology the corners, edges, and faces of objects are constrained to retain fixed relationships permitting rotations and other viewing manipulations thereby. By attribute topology the descriptions of objects are also linked to such attributes as texture, color, weight, and the like.

PREFACE

The ARChitect's Computer Graphics AID (ARCAID) is a system of hardware and software for aiding the architect in design. It is ultimately envisioned as a working environment which permits the completed description of a building without use of paper. This concept challenges us to view architectural design, and, in turn, ARCAID in its full complexity.

This paper is an extension of Space-Form—Computer-aided Design for Architecture (September, 1968) by Max J. Smith, Stephen L. Macdonald, and C. Stephen Carr. [1] That previous report and the present one are the result of a series of studies and meetings extending over a two and one-half year period. Contributions to this project were made by C. Stephen Carr, David A. Luther, Jay A. Schadel, Edward F. Smith, and Max J. Smith. Henry Christiansen, Otto Davidson, Donald Mayhew and Farrin West have been involved with implementing various parts of a structural design system; David A. Luther has implemented an electrical design system; and Edward F. Smith, a system for heating design. Stanley W. Crawley is developing a computer graphics system for radiation shielding design. Gustaaf F. Brest van Kempen reviewed and made valuable comments on the text.

We wanted to view architecture in its full complexity and to take advantage of state-of-the-art developments for both architecture and computer science. We elected to devote a long time to planning the system on the hunch that the subsequent programming effort would be greatly reduced thereby. The programming is now proceeding under the

direction of C. Stephen Carr, and some implementation is available. A few of our projections have already been revised in light of actual experience with the system in operation, and we expect that others will be changed. We wanted a system both comprehensive and economical and devoted our efforts toward an optimal balance between the two.

The present paper is a description of the system in a highly developed form and includes many operations which will not be available at the outset. Based on the present overall description we have already set, and will continue to set, priorities for those operations which are to be programmed at the outset and those to be deferred.

As architects we were interested in the application of computer graphics to building design. With certain changes in terminology the object and attribute processes of ARCAID are also suitable for the design of products other than buildings. Without changes the general-purpose processes are useful for any problem solving which involves words, numbers, and pictures—in short, for nearly any problem solving whatsoever.

ARCAID is a comprehensive system and, as such, a large undertaking. We intend to develop the underlying data structure and to implement a sufficient number of the elements and manipulations specified herein to demonstrate the feasibility of the computer graphics system. As of this writing substantial progress has been made toward that goal.

Dr. David C. Evans
Principal Investigator, ARPA Research Order 829

Professor Stephen L. Macdonald
Principal Investigator, Architecture Research

Dr. Robert Wehrli
Co-Principal Investigator, Architecture Research

CHAPTER I

ARCAID

The ARChitect's Computer Graphics AID

Architectural design is the man-machine development of the description for a building which is sufficient to construct the building. The description is called a scheme.

Design proceeds from the designer's diffuse and unstructured ideas about the scheme to his detailed specification for it. This development moves from the simple to the complex, where "simple" means few objects are incorporated in the scheme, and "complex" means many are. Design also moves from the abstract to the concrete, where "abstract" means few attributes are associated with the scheme and "concrete" means many are.

Typically the designer begins with a consideration of the site and then concentrates on the enclosure/space-use or structural systems. For purposes of this initial discussion it is presumed that he deals first with objects: floors, walls, and roofs for the enclosure/space-use system; or beams and columns for the structural system. Later he considers the attributes of these objects. When a tentative enclosure or structural scheme has been generated, mechanical, electrical, communications, and appurtenance systems might be incorporated.

Chart 1 reveals the structure of the proposed computer graphics system and can be used as a guide to the present discussion. Paragraph numbers correspond to numbers on the chart. The architectural designer at the right of the chart is designing a scheme for a room. He has three walls and the floor in place and is placing the fourth wall and the ceiling. His hand movements with peripheral devices effect manipulations of the new

wall relative to the existing walls and the floor. The picture at the CRT is a graphic representation of the scheme in terms of its points, lines, and surfaces. As the work progresses, words and numbers are included in the picture.

To some degree Chart 1 represents the data structure of ARCAID, but this is not its primary use. Its main purpose is to bridge between this data structure and a systems approach to building design. As such it is meant to reveal to the designer the workings of the computer graphics system and its capabilities.

A system is any group of elements which has or might be conceived as having a high density of internal relationships. In general the internal relationships are more dense than the external relationships between the system and adjacent systems. The smallest element of interest to the designer is the part; between the part and the system is the component—an element made up of two or more parts. In other contexts subassemblies, assemblies and subsystems are elements intermediate between the component and the system.

ARCAID is a computer graphics system which encompasses the architectural designer and his tools. In addition, buildings are treated as [2] systems comprising object and attribute subsystems. To keep this clear ARCAID will always be used to designate the computer graphics system; reference to the building systems will always include the word "building"; and the object and attribute systems discussed will always refer to the objects and attributes of buildings. (For a more intensive treatment of the building systems concept, refer to Appendix A.)

In the following discussion, the second (and third) digit of the

8 OBJECT SYSTEM PROCESS
FORMAT: data, procedures

SITE
CURB
ROADWAY
CIRCULATION SYSTEM
TRAFFIC SIMULATION
ARCHITECTURAL DESIGNER'S VERSION
ARCHITECTURAL FIRM'S VERSION
PROFESSION-WIDE VERSION

STRUCTURE
CONNECTION
BEAM
FRAME
BEAM ANALYSIS
FRAME ANALYSIS

9

ENCLOSURE
SILL
WINDOW
WALL
MINIMIZE PERIMETER WALL
MINIMIZE CIRCULATION SPACE

SPACE USE
CONVERSATION PIT
LIVING ROOM
HOUSE
PLAN ROOMS AND SPACES
PACK ROOMS AND SPACES

MECHANICAL
REGISTER
DUCTWORK
HEATING SYSTEM
RUN DUCTWORK
RUN PIPING

ELECTRICAL
LAMP
LIGHTING FIXTURE
LIGHTING SYSTEM
CIRCUIT WIRING
RUN WIRING
RUN CONDUIT

COMMUNICATIONS
SPEAKER
BRANCH
INTERCOM SYSTEM
CIRCUIT WIRING
RUN WIRING
RUN CONDUIT

APPURTENANCES
CARPET
FLOOR COVERING
INTERIOR DECORATION

6 GENERAL PURPOSE PROCESS
FORMAT: data, procedures

LIBRARY

TEXT EDITING
LETTERS
WORDS
SENTENCES

10

3 PROJECT FILE (for a building scheme)

USE GENERAL PURPOSE PROCESSES TO BUILD-UP OR MODIFY AN OBJECT SYSTEM PROCESS

7 USE GENERAL PURPOSE PROCESSES TO BUILD-UP OR MODIFY A SCHEME OR FOR NON-BUILDING PROBLEMS

USE GENERAL PURPOSE PROCESSES TO BUILD UP OR MODIFY AN ATTRIBUTE SYSTEM PROCESS

SIMULATION

COMPUTATION
DIGITS
NUMBERS
EXPRESSIONS

DIAGNOSTICS

SPACEFORM
PRIMITIVES
BASIC SPACEFORMS
SHAPED SPACEFORMS
OBJECTS
HOUSEKEEPING MANIPULATIONS
SHAPING
ASSEMBLING
VIEWING
ORIENTING

numerical control compiler & processor → PART COMPONENT BUILDING → **5** NUMERICAL CONTROL

plotter compiler & processor → STATIC PICTURE → **4** PLOTTER

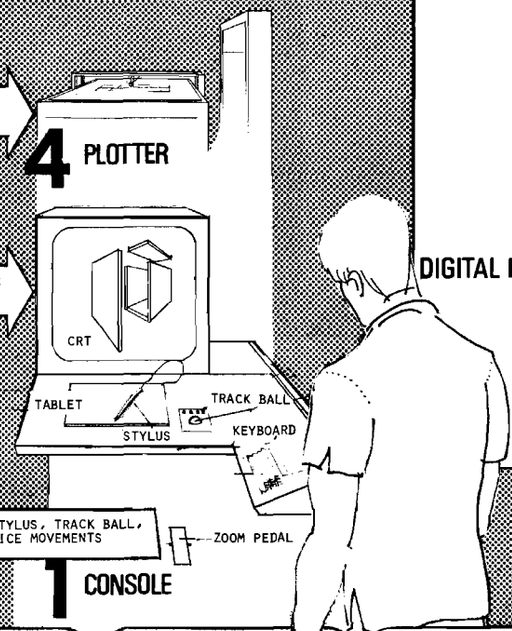
display file compiler & processor → DYNAMIC PICTURE

VIEWING TRANSFORMATIONS

SHAPE AND ASSEMBLE PRIMITIVES, SPACEFORMS, AND OBJECTS TO BUILD-UP AND MODIFY A SCHEME

TYPING AND STYLUS, TRACK BALL, AND ZOOM DEVICE MOVEMENTS

ZOOM PEDAL



1 CONSOLE

DIGITAL PDP8

UNIVAC 1108

2 COMPUTERS

11 ATTRIBUTE SYSTEM PROCESS
FORMAT: procedures, routines, subroutines

AREA
ROOM AREA COMPUTATION
BUILDING AREA COMPUTATION
SITE AREA COMPUTATION

VOLUME
ROOM VOLUME COMPUTATION
BUILDING VOLUME COMPUTATION

COST
UNIT AND PART COST
COMPONENT COST
BUILDING COST

MATERIAL
BILL OF MATERIALS

VISUAL
HUE
COLOR
LIGHT

ACOUSTICS

THERMAL

SAFETY

etc.

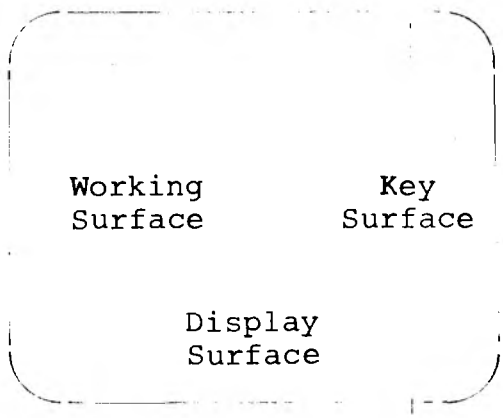
ARCAID - THE ARCHITECT'S COMPUTER GRAPHICS AID

section number corresponds with the chart numbers. For example, Section 1.4 treats chart item 4, The Plotter.

1.1 The On-line Console

We assume that the designer works at a console based upon the cathode ray tube (CRT) as the main source of feedback from the computer. The console has a standard typing keyboard, a tablet and stylus, a track ball, and a zoom pedal. The keyboard permits direct interaction with the computer for numeric and textual inputs and for certain other commands. The stylus and tablet are used by the designer to identify elements at the working surface and move them about. The track ball is used for three-axis rotations of displayed objects and for panning and scanning. The zoom pedal changes the picture to represent moving toward or away from it.

The face of the display is divided into two portions. Keys, other symbols, and text are displayed in the key surface at the right-hand side; and the scheme, or other main picture, is displayed in the central and left-hand side. This area is called the working surface. Together, the key and working surfaces comprise the display surface.[1] See Drawing 1.1.



1.1

The light pen and the mouse are alternative devices for the stylus. Essentially they are used to identify the "present" location of the electron stream at a tracking cross, which is displayed as a bright spot,

[1] A display plane, defined as coplanar with the display surface, can be conceptualized as one of the many planes associated with things pictured.

a cross, or an asterisk. After the tracking cross has been identified, or "located," it can be moved about the display surface.

The light pen may be connected by coaxial cable, fiber optics, or other electronic means, to the console and is used in contact with the working surface of the CRT. This requires the user to hold his arm up to reach the display. In addition the cable is cumbersome to handle, and the light pen has a slow response time. The mouse is a rounded device which fits under the palm of the hand as it rests on an armrest or other surface. Two wheels on the underside of the mouse are connected to potentiometers which, in turn, are connected to the computer circuitry which directs the electron gun. Movement of the wheels manipulates the tracking cross up and down, or from side to side of the display.

Rather than working at the display with the light pen, or at the armrest with the mouse, the stylus works in conjunction with a tablet which is simply a small drawing board containing electronic circuitry connected to the computer. Positioning the stylus on the tablet activates the tracking cross at a corresponding position on the working surface. Existing drawings can be taped down to the tablet and traced with the stylus--a capability not available with either the light pen or mouse.

1.2 The Interfaced Computer

At the University of Utah ARCAID was first implemented on an Industrial Displays Incorporated (IDI) CRT. It was interfaced with a Digital Equipment Corporation (DEC) PDP-8, a small computer sufficient to refresh the picture at the display. A much larger computer, the UNIVAC 1108, was connected to the PDP-8 and used for complex procedures. The software has now been conveyed to a DEC PDP-10, and the research is developing on this

time-sharing machine. Other computers may be used, but in any event the system is planned for time-sharing where several consoles are remote from a large computer. A large number of schemes can be designed concurrently by several designers in such a time-sharing system.

1.3 The Project File

Design can be characterized as the building up of the description (or scheme) for a building in a project file. This file is essentially storage. It is analogous to a set of blank tracing sheets which the designer has numbered and set aside for working drawings. Each building to be designed is designated as a separate project requiring a separate project file. The sheets are perfectly "erasable", permitting the designer to make his first schematic sketches and then erase and change while working continuously toward the final degree of detail needed for construction documents.

As it is built up and modified, the scheme is stored automatically in its "present" state, i.e., with as much detail and as many attributes as have been developed or assigned by the designer. The scheme-as-stored does not affect the basic arrangement of the data structure of ARCAID; neither is the scheme-as-stored affected by those transformations of its data which are continuously needed for viewing manipulations and the like.

1.4 The Plotter

Traditionally, architectural schemes have been described with two-dimensional drawings. For schematic and preliminary design, perspectives are often used; and for working drawings, orthographic projections. The

plotter can produce such drawings on translucent tracing cloth or paper, which can then be used for printing additional copies. The plotter has a turret which holds a number of pens. The turret moves to and fro on an arm for "X-X" lines, and the arm moves back and forth across the drafting surface for "Y-Y" lines. The plotter is driven by the same scheme information in the project file which drives the CRT but requires a different compiler and processor.

As time goes by such hard copy may be obviated by using CRT's at the construction site. The contractor will "dial in" to a central computer for the scheme of the building he is constructing, and he will be able to access not just that information which the architect might have presented on working drawings but to any information whatsoever that has been topologically incorporated in the scheme.

1.5 Numeric Control Machine

Presently industry is using numeric control computer programs to drive machine tools. Such methods have not been quite so common for building construction at the job site, but the advent of prefabrication especially for the type proposed by Moshe Safdie in his Expo '67 Habitat, shows promise for such automated construction. Safdie used a more or less standard concrete box as a living unit and stacked the boxes in a random "pile" to form his apartment house. Just as for plotters, numeric control machines can be driven by the same information in the project file which produces the CRT pictures, but a different compiler and processor may be needed as shown in Chart 1.

1.6 General Purpose Processes[1]

In addition to the project file described briefly above, ARCAID consists internally of a set of general-purpose processes (refer to Number 6 on Chart 1), a set of building systems processes (Numbers 8 and 11), a set of automatic data transformation devices, and the library. The three essential general-purpose processes are text-editing, computation, and graphics. They permit manipulations of three main element types: words, numbers, and pictures. In addition, the chart shows others--simulation and diagnostics--which might be added at a later time.

For text-editing and computation, existing formal languages will be employed at first. For example, COBOL, FORTRAN, and ALGOL. As improved, higher level languages become available they will be added. For graphics, we propose here the SPACEFORM language, which is described in later chapters. SPACEFORM includes not only picture-making but associated text manipulation as well.

Every process consists of data (elements to be manipulated) and procedures for manipulating this data. The data may be stored in advance in connection with one of the processes or input as needed by the designer. For example, in SPACEFORM, a number of basic spaceforms are stored in the system. These are the cuboid, hexagonal rod, triangular rod, cylinder, and sphere. This stored data can be augmented by the designer, who can build-up new spaceforms such as the tetrahedron, ellipsoid, pyramid, and the like. He may also build up complex spaceforms representing a building component, such as a wall or floor. He

[1] Here the word *process* takes on a particular computer science definition; i.e., computer data together with computer procedures for acting upon the data.

may store and retrieve these just as he does the basic spaceforms. A building scheme in the project file is built up of pre-stored data from processes, together with new data input by the designer.

1.7 Uses of the General-purpose Processes

The general-purpose processes can be used not only for the design of building schemes, but for any problem involving words, numbers and pictures. In short, any problem whatever. They may be used to build up or modify one of the object or attribute systems or to program a new procedure.

1.8 Object System Processes

Building system processes consist of associated computer graphic procedures and data which aid the designer in accomplishing a given task. [3] General-purpose processes permit the use of words, numbers, and pictures in a broad range of problems or activities. Object and attribute processes aid in more specific design tasks with particular relevance to architectural design.

For designing buildings there are two types of processes: object system and attribute system. For any given building there is a small and exhaustive set of object systems. In ARCAID the processes for dealing with object systems are as follows: site, structure, enclosure/space-use, mechanical, electrical, communications, and appurtenances.

1.9 Communication among Object System Processes

All of the object system processes are arranged side by side in Chart 1. This offers the advantage of comparing them for common procedures. In electrical design, for example, wiring and conduit routing algorithms are proposed. They work as follows [4]: when the designer

completes his enclosure/space-use scheme for a building, he lays out in plan the needed electrical outlets and then commands an algorithm to produce automatically a routing of circuits, wires, and conduit. He can then interact with the system for whatever changes he wants.

It is likely that these procedures for electrical work are also suitable, with minor variations, for communications. By the same token they can be extended for the routing of piping and ductwork in the mechanical system. In this way many fewer procedures are needed for completing ARCAID than if each building system were considered separately without thought about the others.

At the left of the string of object system processes (in Chart 1) is a space referred to as a "format." This serves not only as a heading for the data and procedures of the object system processes, but also to indicate that an analysis of the various building systems and their design may result in other procedures which are general rather than specific in their application.

It is difficult to predict how much interaction between object system processes will be needed by the designer using computer graphics. Under present practice the various building systems are often designed by independent persons or teams located in separate offices or even in different cities. Enclosure/space-use is often designed by one team, and structural, mechanical, and electrical systems by other teams.

It is expected that there will be greater need for quick access to the procedures of the various building systems than for access to the data of these systems. Thus it is fortunate that the procedures require a relatively small computer storage capacity as compared with the data. It seems reasonable that the perfectly coordinated, or synthetic, design system be based on an ability to access quickly both data and

procedures; and this will come as techniques are improved and costs drop. For example, consider analytic procedures traditionally used for structural design. The designer begins with an approximate configuration for the structural system, assigning by experience, or rule-of-thumb, approximate spans, member sizes, joint types, and the like. He then determines by analysis whether or not his estimated configuration is properly stressed. If it is not, he reassigns the values and analyzes the structure again until satisfactory. But how does he arrive at the original configuration? Only by making assumptions and constraining other building systems in advance. In the perfectly synthetic design environment, however, all these building systems would be responsive to changes in the others. Such an environment can be achieved with ARCAID.

1.10 Communication between Object System Processes and the Project File

The scheme in the project file is the main link between data and procedures of the object and attribute systems. The data of the scheme is built up in a way best conceptualized as a tree. The building itself, for example, occupies the top node of the tree; at the next lower level are rooms and spaces consisting of floors, walls, and roofs or ceilings; and these latter elements are at a still lower level. In some cases the machine descriptions of rooms and spaces and, in turn, of the floors, walls, and roofs or ceilings are incorporated in the scheme itself. In other cases there is no incorporated description but only a "pointer" to some appropriate data in an object system. For example, the windows of all manufacturers will ultimately be stored topologically as data in the enclosure system. The windows of a particular manufacturer are associated with a given building scheme by pointers which serve to access the window information when needed for display or plotting and the like.

1.11 Attribute System Processes

Since the attributes of objects are as broad as human experience itself, ultimately a great many attribute systems are possible. For the present, however, only processes for the following attributes are contemplated: area, volume, cost, materials, visual, acoustics, thermal, and safety. Like the object system processes, the attribute system processes consist of data and procedures, but there tends to be less data for any attribute system than for an object system. This is so because the attributes for a particular object—say, a certain type window of a particular manufacturer—are associated or stored with that type of window as data in the enclosure process. The attributes of the window might be dimensions, frame material, glazing, operation type, finish, cost, and the like. It is the purpose of the attribute processes generally to collect, sum, or otherwise deal with these attributes acting "through" the scheme in the project file. Suppose that the designer produces the scheme for a small building having three windows in each of two window types. As he positions the windows in the picture of the building shown on the CRT, their attributes are associated with the tree structure of the scheme. Window types, numbers, and dimensions can then be accessed by the area (attribute) process; the procedures in this process multiply widths and heights of the six windows for a total window area. When appropriately queried, the individual and total window areas are then displayed on the working surface of the CRT.

1.12 Communication among Attribute System Processes

The attribute system processes support one another. In particular, the attribute systems at the left of the chart support those at the center and right of the chart. In the paragraph above, it was shown for

windows how the area process automatically accesses data from the object processes via the project scheme. This is done for all items associated with the scheme according to architecting[1] conventions for such area computations. The same applies, of course, to volume computations. The conventions permit use of computed areas and volumes as needed for the other attribute systems. For example, wall areas for a building are available in the area process. They can be accessed by the cost process and multiplied by unit costs for walls to provide the total cost of walls for the building. All other elements of the scheme can likewise be multiplied by their part or unit costs for an overall estimated cost for the building. By the same token, the wall areas can be used to produce a bill of materials for walls in the materials process, to calculate light reflection in the visual process, to determine sound absorptions in the acoustics process and heat loss in the thermal process, and so forth.

1.13 Communication between the Attribute System Processes and the Project File

It has now been described generally how the designer builds up and modifies a scheme for a building in the project file. Elements for the building scheme are either stored directly in the project file or accessed by pointers to elements "permanently" stored as data in one of the object system processes. Data associated with the scheme are in turn accessed by the general-purpose, object, or attribute system pro-

[1] "Architecting" refers only to the activities of an architect, and is, therefore, more specific than the traditional word "architectural," which refers to these activities as well as to the history, philosophy, attitudes, and buildings of the architects.

cedures. The attribute system processes access appropriate data, execute procedures, or when commanded by the designer provide desired sums or combinations. Another example of this is the color system. In designing a building the designer may assign the values for certain colors and not for others. He may assign the bathtub as blue and the heating registers as bronze without assigning any other colors. When the building scheme is nearly finished, he may wish to complete the color system for the building. He then summons from the visual system process the color list, which includes all those objects for which a color must be selected giving values for those colors which have already been selected and leaving blanks for those which have not. As the designer makes and enters his selections, they form a portion of the scheme which is then printed out as a painting schedule and sent to the painting and decorating contractor.

In the initial paragraphs of this chapter it was proposed tentatively that the designer begin with objects first and then concern himself with attributes. Now we can see that the reverse works equally well, for he may assign the desired attributes of certain objects at the outset. These attributes then constitute constraints on the objects selected or designed at a later time.

1.14 The ARCAID Library

Words, numbers, and graphic elements are stored as data with the general-purpose processes. All the objects used in buildings are catalogued and stored in applicable object system processes together with the attributes which refer to each of the objects. Data required in connection with procedures for dealing with attributes are stored in appropriate attribute system processes. All other information whatsoever that is needed for ARCAID is stored in the ARCAID library.

CHAPTER II

The SPACEFORM Graphics Language

2.1 The Rationale for Spaceforms

ARCAID aims for techniques which permit free-flowing interchange of information between the designer and the computer. This real-time conversation is essential to merging the best qualities of the machine (infallible memory, accuracy, and speed for doing routine problems) with those of the designer (creative imagination, curiosity, intuition, and invention).

The display and stylus will replace the drawing board and pencil. The typewriter keyboard and other peripheral devices will assist in the communication.

The purpose of this section of the paper is to describe these movements in greater detail with particular reference to the SPACEFORM graphics language. ARCAID may be thought of as a collection of languages —at least one language for each object and attribute system. The SPACEFORM graphics language may be used for problems which are otherwise independent of ARCAID; but when it is used for architectural problems, SPACEFORM is interconnected with all the other languages.

Take structures for example. Structural languages such as FRAN, STRESS, STRUDL, [5] SAMIS, and FRAME have been available for some time; and two of them, FRAME and SAMIS, will be implemented in the structures process of ARCAID. The analytic calculations are done by proven, rigorous methods. A primary difficulty with them, however, is inputting the original description of the structure and then interpreting the usually massive and sometimes incomprehensible output.

The input has necessarily been by x, y, z numeric descriptions laboriously punched on cards, or at best by typing. The conversion from the iconic drawing of the structure to x, y, z coordinates for each joint and member is foreign to the architect and cumbersome for the engineer.

SPACEFORM, however, permits graphic input not only for the structural members but also for superimposed loadings, joint fixedness, and the like.

By the same token, it has been difficult to pore through long lists of numeric output for a certain calculation. Here, again, output can now be graphic, perhaps showing exaggerated deflections in the structure, or showing stresses alongside critical members, and the like.

The basic idea of SPACEFORM is [6] the use of a set of perspective building blocks called "spaceforms." These are three-dimensionally described graphic elements which can be displayed two-dimensionally on the CRT. Spaceforms can be built up and modified in various ways or cut and otherwise shaped for almost any configuration. They can be assembled in any number of ways.

The rationale for spaceforms over a more conventional drafting approach is that the latter is neither available nor appropriate in computer graphics. As previously stated, only the stylus and its tablet afford the tracing of lines in which the vertical and horizontal, for example, could be maintained by drafting instruments. Even so, inaccuracies are likely, and the tablet and working surface are much smaller (10 to 17 inches square) than the drawing sheets architects are accustomed to.

But whereas the relationships in a drawing cannot readily be input by the stylus, light pen, or mouse for each occurrence, they can be

input topologically in advance of their use. The computer maintains the needed relationships—such as keeping the edges at right angles for a rectangle or the faces at right angles for a cuboid.

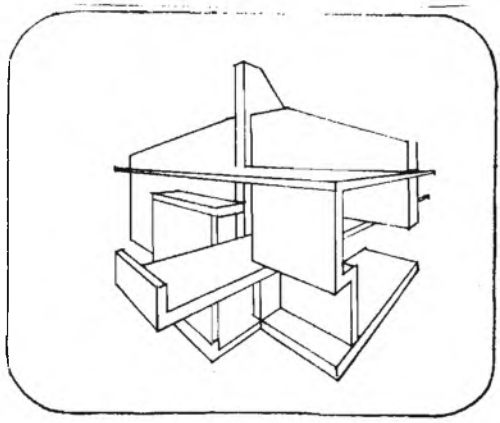
Spaceforms, then, are the main elements of design. They are simple yet sufficiently complex to incorporate a number of topologically constrained relationships. More important, the objects which represent the final result of the design all occupy space (they have volume), and this suggests that basic drawing elements should also be three-dimensional in nature instead of volumeless lines and points. And at a display we can manipulate spaceforms in a manner analogous to constructing objects in the real world.

The basic spaceforms are composed of primitives—points, lines, angles, curves, and surfaces. These primitives are not the main elements of the graphics language, but they are accessible for building up or modifying new spaceforms or for several other purposes. The basic spaceforms available first are the cuboid, hexagonal rod, triangular rod, cylinder, and sphere.

These basic spaceforms are simply three-dimensionally described graphic elements made up of points, lines, curves, and surfaces. When incorporated in the spaceform, points become corners, lines become edges, and surfaces become faces. When sized, cut, or otherwise shaped, basic spaceforms become shaped spaceforms. Basic spaceforms have only shape, unit dimensions, and volume. Shaped spaceforms have dimensions and are more complex than basic spaceforms. When still more attributes are assigned, or when so-named, spaceforms become "objects"—the parts, components, and systems of buildings. In computer graphics, images of objects rather than the objects themselves are manipulated. It is convenient, however, to speak of the images as if they were the objects themselves, and this convention

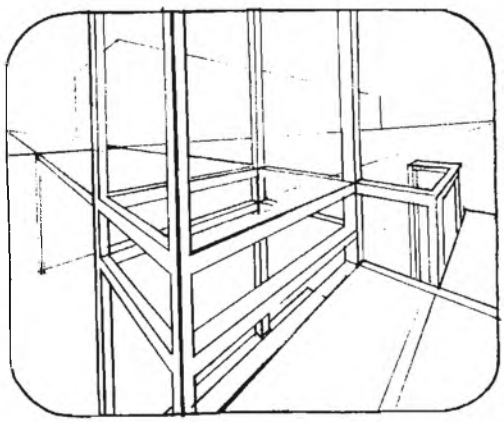
is hereby adopted.

The designer uses primitives to make new basic spaceforms. He uses the latter for shaped spaceforms and the new shaped spaceforms to build objects. Both basic and shaped spaceforms are stored in the library of the SPACEFORM language for later use, and the objects are stored in conjunction with appropriate object systems.

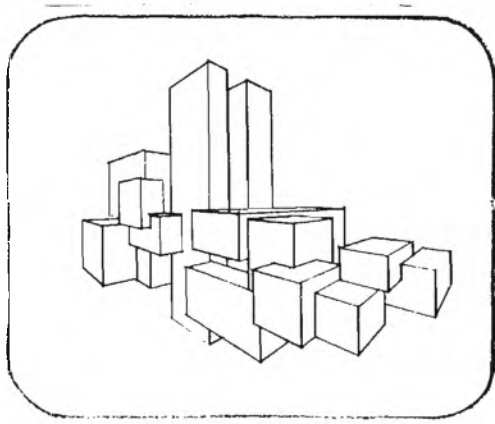


2.1

Buildings may be assembled in a number of ways as shown in the drawings. In Drawing 2.1 flat (cuboid) spaceforms are assembled as the enclosure elements—floors, walls, and roofs—of a building scheme. In Drawing 2.2 slender (cuboid) spaceforms represent the beams and columns of a frame (which is appropriate for a building in which the structural elements dominate the designer's first considerations); and squarish (cuboid) spaceforms represent rooms and spaces of the building scheme in Drawing 2.3. (The corners of the illustrations are curved to indicate the outline of a CRT face; but this border is omitted when the illustration represents the real world.)



2.2

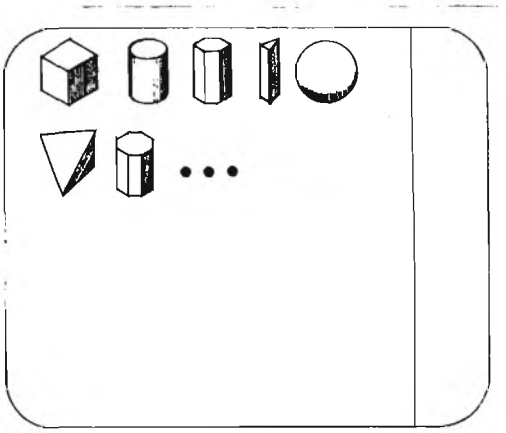


2.3

There are two modes for design—the abstract and the concrete. In the abstract

mode the designer uses the SPACEFORM language with its primitives and shaped spaceforms. He treats the scheme essentially in geometric terms of its extent and shape. If, as in Chart 1, he manipulates an element into position, that element is treated by the computer as a cuboid even though the designer may *think* of it as a wall, floor, or ceiling. At a later time he can name the cuboid "wall," whereupon it will take on those attributes of walls stored in the enclosure system data bank; or he can assign the cuboid a particular wall type as selected from the enclosure system data bank. In this latter case the cuboid takes on not only the attributes of walls generally but also those attributes of a specific wall type.

Drawing 2.4 shows typical basic spaceforms available for abstract mode design. The top five are those initially available in the SPACEFORM language data bank, and below these are two which might subsequently be built up by the designer.



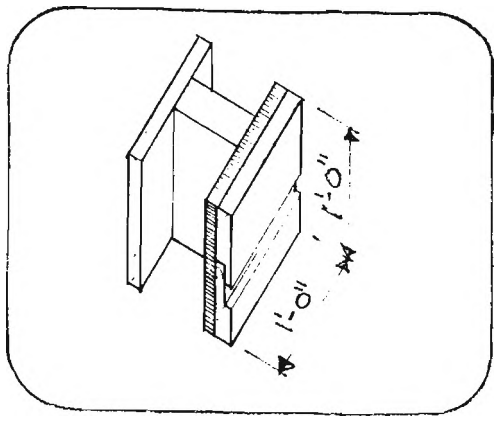
2.4

As an alternative, the architectural designer may work in the concrete mode, treating the same wall or ceiling directly in concrete terms. Instead of using the cuboid of the SPACEFORM language as the basic element, the designer may use a "square-foot-through-the-wall." This unit can be selected from many such units available in the enclosure data bank and placed in the key space so as to be available for use when needed. For example, a unit of exterior frame construction might consist of a square foot of exterior paint, cedar siding, insulation board, a portion of a stud, a square foot of gypsum board, and interior paint. A copy of this unit as shown in Drawing 2.5 can be translated from the key

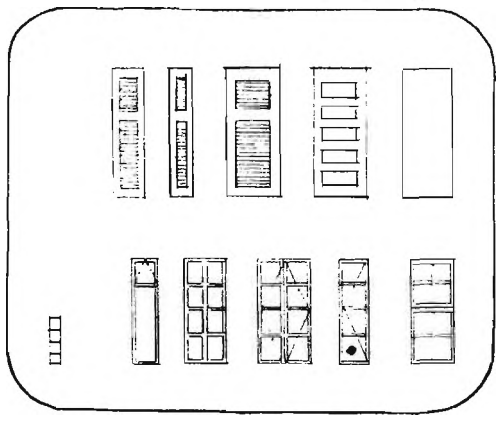
space, sized, and assembled in a building scheme. In this case, of course, it already bears the attributes of the selected wall type.

Shown in Drawing 2.6 are windows and doors which, like the typical square-foot-through-the-wall, might be accessed from the enclosure data bank for use in the concrete development of a scheme. The SPACEFORM language may have been used earlier to build up this library of windows and doors; when additional ones are needed, they can be built up in a similar fashion.

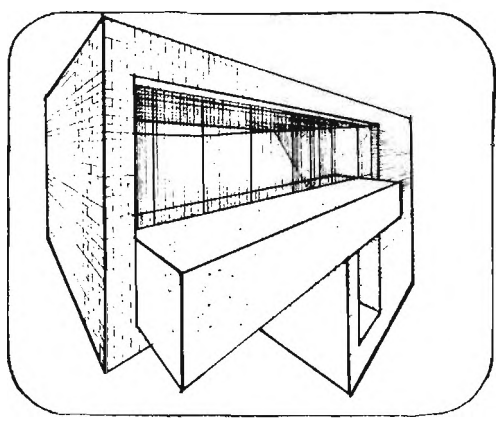
Drawing 2.7 shows a building with masonry joints and cement or stucco texturing—a degree of detail which may be possible through later improvements on the computer graphics system. At the stage of development in Drawing 2.7 it is not possible to determine whether the designer worked initially in the abstract or concrete mode.



2.5



2.6



2.7

2.2 Topology

The above discussion of spaceforms, objects, and building schemes mentions topology, which refers herein to relationships maintained by the computer. These relationships include the geometric but extend beyond it to other relationships within and between objects and their attributes and values.

2.3 Perspectives

Requirements for computer graphic perspectives differ from those for the customary perspective drawings of architects. For the latter, [7] certain simplifying characteristics tend to hold. The buildings (or objects) pictured are rectilinear and consist essentially of two families of parallel planes; each perspective is considered singly, and the station point is located about five feet above the horizon and opposite one vertical edge of the building so as to display favorably two adjacent building facades. On one facade one family of parallel planes is major and the second family minor. These families are reversed for the second facade. For such buildings two or at most three vanishing points are required.

To exploit the capabilities of computer graphics, however, one is not limited to rectilinear objects; multiple perspectives represent motion, and the station point for any one perspective may be at any location relative to the object. What is needed, then, is a method which determines on the basis of the present position of any one face of a complex object the shape of its image on the display surface. Transformations by [8] matrix algebra are used and can be performed by software or hardware. These transformations employ matrix algebra to convert from x, y, and z object descriptions in the project file to two-

dimensional descriptions for the display.

2.4 Hierarchical Organization

The ARCAID computer graphics system is organized hierarchically with the largest, most general elements at the top, or trunk of the tree, and lesser elements lower at branches or leaves. This is further described and illustrated in Chapter 3.

This same organization applies to the schemes of buildings or other objects stored in the project file. Here the building itself is the major item at the top of the tree. At the next level might be roofs, floors, and walls. At a still lower level, under walls, might be doors and windows, and under windows might be jambs, sills, heads, etc. This organization ensures that all parts of the building description are logically and topologically related to one another so the CRT pictures appear with appropriate relationships.

2.5 Associative Memory

For ARCAID an associative memory provides high-speed data searches based on the content or subject matter of the data rather than specific "addresses" in the machine. Associative triplets are employed [9]; that is, objects and their attributes are joined in threes. For example, we may use the triplet: the *material* of the *window* is *wood.* Here *window* represents the object and *material*, one attribute of the window. *Wood* is the value given the material of the window. The statement is only one of many that could be made about materials, windows, and wood. Other, more general statements are:

> What are the various attributes, such as
>
> *material*, for windows?

What are the types of *windows* stored?

What types of materials, such as *wood*, do the windows have?

In a still more general way, one can query:

What are the various *attributes*, like *material*, for all objects?

What are the many *objects* stored—windows and others?

What are all the values for attributes—wood, steel, blue, red, etc.?

In this context the terms attribute, object, and value (unlike their use in the context of building systems) have a syntactic, rather than a semantic relationship. To illustrate, one can make the statement: the *color* of the *wood* is *yellow*. Here *wood*, which was the value of the previous statement, is now the object. *Color* is the attribute and *yellow* its value. To take this one step further, one may say: the *shade* of *yellow* is *dark*. Now *yellow*, previously the value, has become the object.

The above statements also show how chaining between triplets can extend indefinitely the search potential of the triplets, for the statements move from windows, to wood, to yellow, to dark yellow.

In addition to the triplets, a scheme is used whereby each word in its context is searched by its spelling. This means that each word must be used unambiguously. If the word *elevation* is used, for example, it must have a single, unambiguous meaning. If elevation represents a height above sea level, it cannot also mean the facade of a building.

CHAPTER III

Housekeeping Manipulations

3.1 Log In

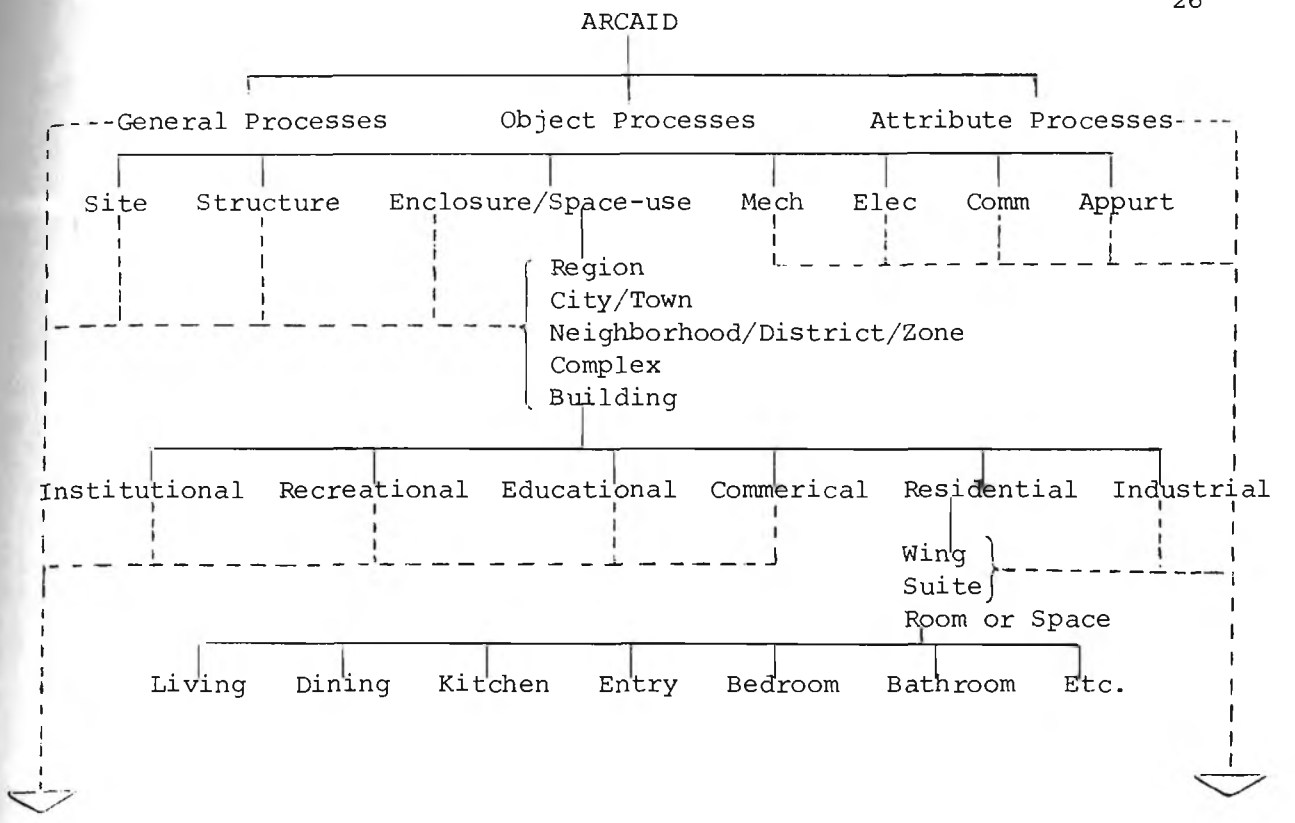
To log in, first activate the console, type ARCAID, and execute. When the list shown at the left below appears on the working surface, type in items as shown in the example at the right. Type in order from top to bottom.

Displayed List	Typing Literal or Movement
ARCAID	
Project Name	Hughes House
Project Number	694703
Designer	Ralph Robbins
Firm	Robbins and Bobbins
Time limit (minimum)	20
Execute	(Press return button.)

ARCAID is organized hierarchically with the system itself as the highest "node," node subheadings at the next lower level, and specific items at the lowest level. This is of course closely allied with the system-component-part breakdown of the systems approach to building design and construction.

The ordering of the tree is as shown in Drawing 3.1. (Only the center branching is shown to exemplify the method. Dotted lines indicate the missing left and right branching, which hopefully the reader can imagine.)

Following the final "execute" of logging in, a listing of general purpose, object, and attribute processes is displayed as shown in



3.1

The ARChitect's Computer Graphics AID

ARCAID

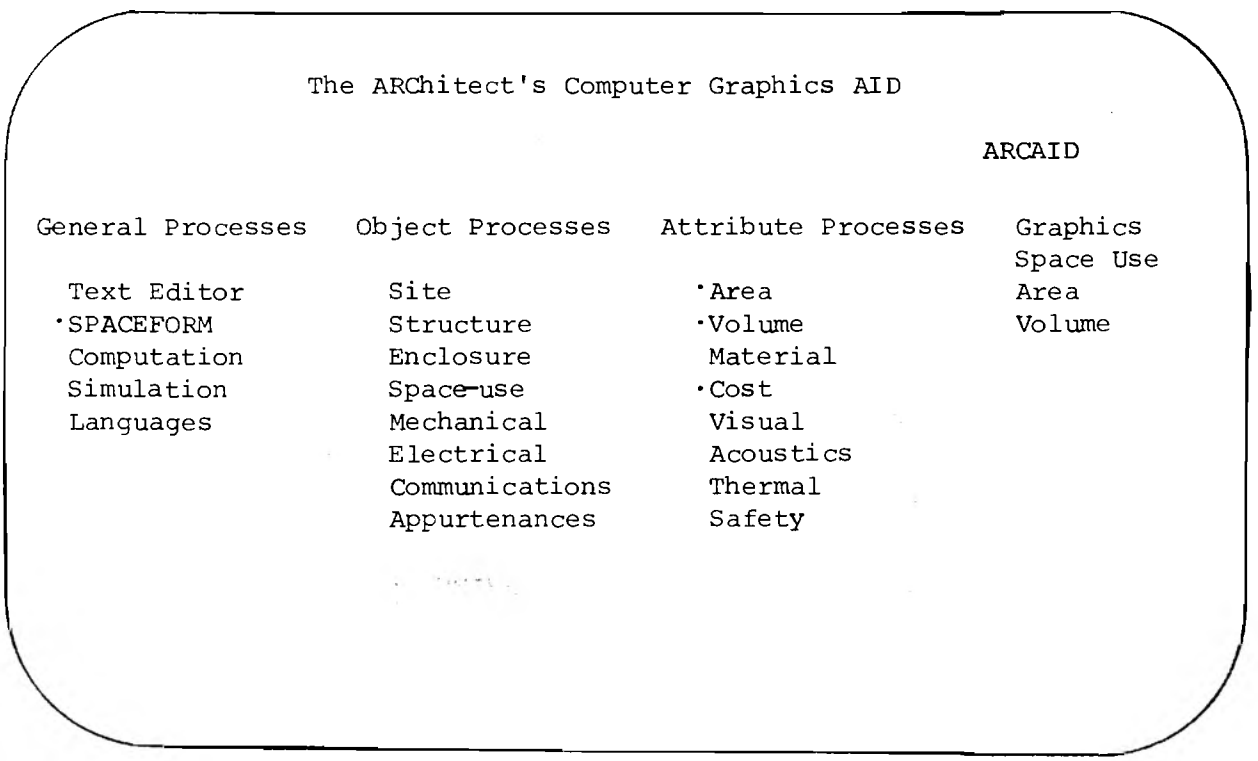
General Processes	Object Processes	Attribute Processes
Text Editor	Site	Area
SPACEFORM	Structure	Volume
Computation	Enclosure	Material
Simulation	Space-use	Cost
Languages	Mechanical	Visual
	Electrical	Acoustics
	Communications	Thermal
	Appurtenances	Safety

3.2

Drawing 3.2. This is the highest level in the tree of data.

To select a process use the stylus to "touch" the name; then a spot from the tracking cross appears following the "execute." Only one of the general purpose processes or one of the object processes may be active at any time, but they may be cycled in and out from time to time during design. As many attribute systems as desired may concurrently intercommunicate with the project file; but, again, only *one* can be active.

Drawing 3.3 shows spots opposite the SPACEFORM language and the Area, Volume, and Cost Processes to indicate that these have been selected for use.



3.3

Given the tree structure, the following manipulations are needed:

1) Moving down through the tree from the general to the specific.

2) Moving up through the tree from the specific to the general.

3) Moving from a node at one level to another at the same level.

4) Scrolling through additional information at a single level, or node, when all information cannot be displayed at once.

In combination, these manipulations permit the designer to move from any point in the data to any other. The manipulations are accomplished as follows:

1) Moving down through the tree is allowed at any time after the first of the various processes has been displayed as shown in Drawing 3.2.

Here the top node, ARCAID, is listed with two lower levels: (1) the process types—general, object, attribute—and (2) the available processes under each type. At lower nodes only two levels are displayed concurrently. If a single attribute process is touched, it alone will be activated and its index displayed. Any general purpose or any object process which is touched will be activated alone and its index displayed along with any general-purpose processes or object processes which have also been touched.

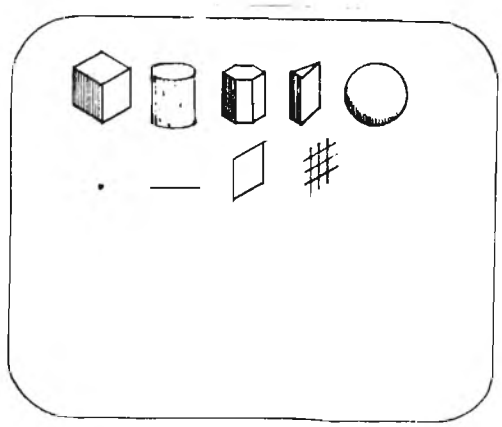
2) All information is displayed in conventional left-to-right, top-to-bottom fashion with a heading at the top or top left. To move up through the tree touch the heading and execute. All items at the same level as the touched heading will be displayed together with their superheading. If the new information is too large for the working surface, the information

begins with the topmost—in alphabetically ordered information, for example, "A" is topmost.

3) To move to another node at the same level, it is necessary to move first to the next higher level as described in (2) and then down to the desired node by touching and executing.

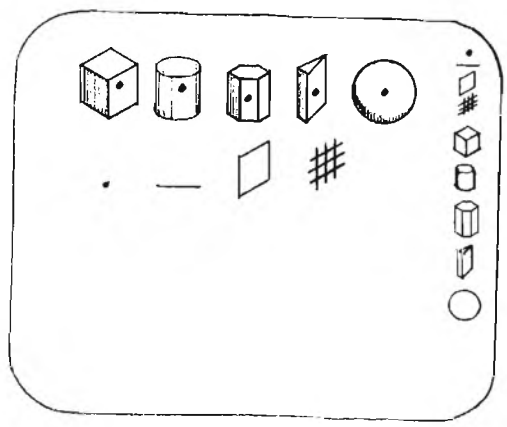
4) When listings become large, there is a 2-D location marked by an asterisk at the top and bottom (or top left and bottom right) of the displayed data. To scroll to information shown above that currently displayed, touch the top (or top-left) asterisk and execute; and to scroll to information below that displayed, touch the bottom (or bottom-right) asterisk and execute.

Using the above description of the tree structure of the data we continue the description of logging in. Assume that the SPACEFORM language has been selected. On execution, the index of spaceforms appears at the display surface as shown in Drawing 3.4. An index includes all the primitives, the five basic spaceforms, and any others which may have been stored by the designer.



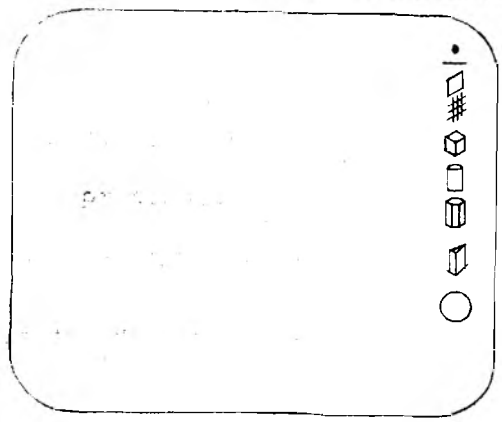
3.4

Again the desired spaceforms are touched as in Drawing 3.5. At the execute command the selected spaceforms appear on the key surface



3.5

and are ready for use in a building scheme. Undesired spaceforms are erased, leaving the working surface clear to receive the scheme as in Drawing 3.6.



3.6

3.2 Touch

To touch is to identify an element which is to be manipulated. Touching is effected by the tracking cross, which is a small cross on the display surface indicating the active location of the light pen, stylus, or mouse. To touch one first points the stylus at the tracking cross, executes, then points to the desired location on the display surface and executes again making sure the tracking cross has been repositioned at the desired location. Spaceforms can be touched unambiguously in three locations—the corner, edge, or face. The various manipulations are referenced to these locations. When a whole spaceform or a cluster of spaceforms is to be manipulated, its name is typed prior to the command for the manipulation.

When an element is touched in the key surface the stylus is "inked" with that element, and as many copies (or instances) of it as desired can then be placed on the working surface. The stylus places additional copies of this element until it is inked with another element or command.

As indicated above, touching is used for placing elements as well as for identifying them.

3.3 Place

As described above, elements are placed on the display surface with

the stylus. It should be noted that all locations on the display surface are essentially 2-D locations, which are relative rather than absolute. For text, the character-to-character and line-to-line (of type) relationships are apparent and hold for printouts of displayed text. However, for objects which have fixed dimensional relationships (either internal or external) it is essential that the designer name these dimensions. For a discussion of this refer to later chapters.

3.4 Execute

The execute command tells the computer that a decision has been made and that the active command should be carried out. To identify or place an element the execute command is accomplished by pressing down on the stylus to activate the switch in its tip. When the teletype is used, however, the execute command is effected when a command is fully typed or when the carriage return is struck.

3.5 Recover

It is possible to recover the last command given and its effects. To do this, type RECOVER and execute. Recover will delete a command in process if given before the final "execute" of such a command.

3.6 Erase

Erasing removes part or all of a picture from the working surface. To erase, touch the elements to be erased, execute, and type the word ERASE, then execute again. If no elements are specifically touched, the entire picture will be erased from the working surface. To avoid losing adjacent elements which one wants to retain, it is best to name as well as touch the elements to be erased.

3.7 Delete

To delete is to remove an element from both the picture and the scheme. Deletions become garbage. To delete, touch the element or elements to be deleted, execute, type DELETE, and execute again. When this is done the element disappears from the display surface and from the scheme. And again, to avoid losing adjacent elements it is best to name as well as touch those elements which are to be deleted.

3.8 Store

To store is to remove an element from the working surface and place it in storage. Storage may be in a general-purpose, object or attribute process, in the ARCAID library, or in the project file. To store in the project file, first touch the element, execute, label, execute, and type STORE. To store elsewhere than in the project file it is necessary to type the storage location before typing the word STORE.

As design progresses, the scheme is developed in the project file. When the designer intends to store his work as part of the scheme he has only to perform the store manipulations. In the early phases of design, there may be more than one alternative scheme under consideration. Any number of alternative schemes can be stored in the project file, but they must be separately named for positive identification and later retrieval.

Further, the designer may want to store some element in one of the ARCAID processes. For example, he may design a window and want to use it in the scheme at hand and also store it along with other enclosure process data for use in a future scheme. The window is designed in the project file, associated with the present scheme, and stored in

the enclosure process by appropriate commands.

3.9 Constrain; Constraints

To constrain is to limit the degrees of freedom available in a scheme. It imposes a decision on the scheme which then affects later decisions.

A *constraint* is a specific storage representation of the relationship between elements. This constraint reduces the number of degrees of freedom of the system.

3.10 Name

To name is to assign text to a graphics element or to give it an identity. The text may be a name or one or more attributes. To name touch the element, execute, type LABEL, and execute again. If the element has already been named, the name appears opposite it. If it has not been named, the words TYPE NAME appear at the display surface. In this case type the desired name or attributes and execute again. The text then appears as a label at the point of stylus contact until the next command is given, at which time the label disappears.

Names may be used to further identify elements. For example cuboids, cylinders, or other spaceforms may be named Cuboid 1, Cuboid 2, etc. Such names can later be labeled and thus serve to distinguish more positively one element from another during retrieval or when the picture is ambiguous.

Naming objects associates them with attributes stored in ARCAID. For example, naming an object "window" associates it with attributes of windows in the enclosure data.

3.11 Label

Labeling is used to display names and attributes. To label an element, touch it, execute, type the word LABEL, and execute again. As discussed above, labeling may be used in addition to touching as a way of identifying a graphic element. Labels and other text are not available in perspective but are only available when graphic elements are shown in orthographic projection.

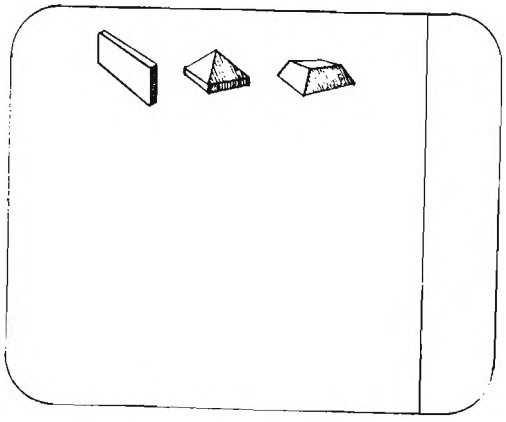
3.12 Duplicate

There are two ways to duplicate an element--making a copy or making an instance. To make a copy touch the displayed element, execute, type COPY, execute, and place the copy as desired. To make an instance follow the above procedure typing INSTANCE instead of COPY. One may also duplicate by touching an element in the key surface and placing it on the working surface. In this case an instance is always made unless a copy is specifically commanded by typing COPY.

Copies and instances appear the same at the display surface. They differ in the way they are referenced in the data structure of ARCAID. Instances are not described directly in the project file but are referenced by a pointer to a description in one of the object system processes. All instances are referenced to the same master description; therefore, a change in the master changes all instances. For example, a window appearing in a scheme may be an instance of a master window in the enclosure system process. If this master window is described as three feet wide, all instances appearing in the scheme as pictured on the display will be three feet wide. A change in the width of the master to four feet changes *all* instances to four feet.

Copies are described directly in the project file. As with the instances they might also be created from a master window, but they are no longer referenced to it by pointers. Each window copy may be altered independently; a change in the master window from three to four feet leaves the scheme windows unchanged. If a general width change from three to four feet is desired, each copy must be changed individually; or each copy may be given a different width.

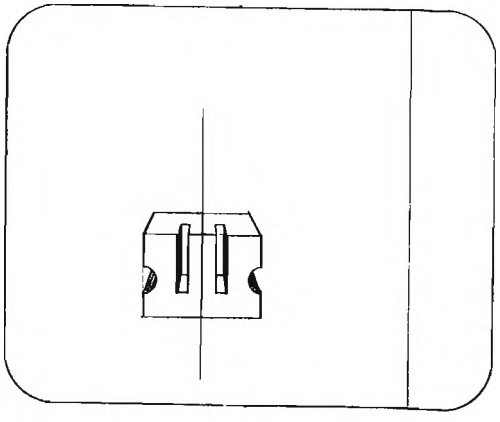
One may create as many copies or instances as desired. Also masters may be made of any element. Drawing 3.7 shows three shaped spaceforms which might be used as masters. These are a thin cuboid, a pyramid on a flattened cuboid, and a truncated pyramid. One may duplicate them by typing COPY or INSTANCE as described above, by placing them in the key for immediate duplication, or by storing them in the index of shaped spaceforms for future duplication.



3.7

3.13 Flop

Flopping permits duplication of graphics elements which are reversed from left to right hand. To flop a graphic element on the working surface touch it, execute, type FLOP, and execute again. The element disappears and then reappears



3.8

flopped. To duplicate flopped elements follow the procedures described in Section 3.12 typing FLOP before the words COPY or INSTANCE. Drawing 3.8 shows a shaped spaceform built from an original half and a flopped half.

3.14 Log Out

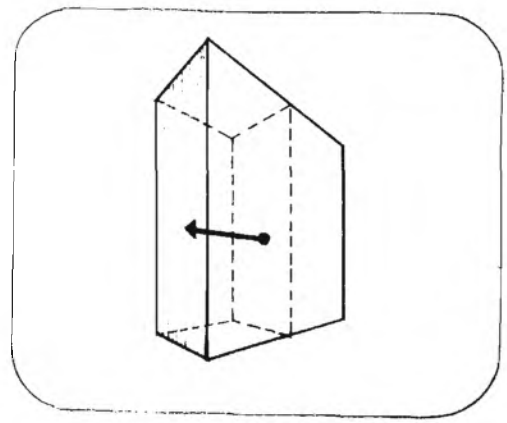
The designer may log out at any time. To do so he types LOG OUT and executes. This clears the working surface, and the message GIVE STORING INSTRUCTIONS is displayed. If the scheme is to be stored, he completes the manipulations; otherwise he repeats the LOG OUT and execute commands. These commands erase the work and turn off the system.

CHAPTER IV

Shaping Manipulations

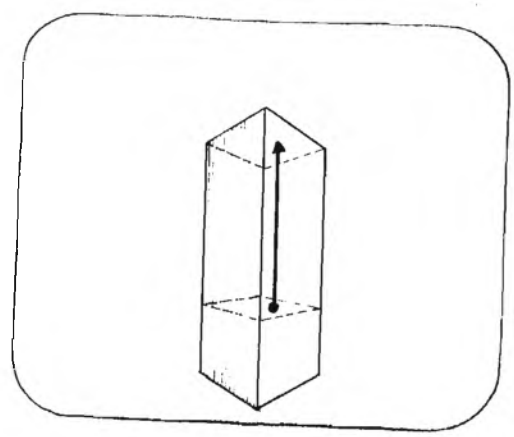
4.1 Contract-Expand

To contract or expand primitives, spaceforms, or building parts, touch the corner, edge, or face which is to be affected and execute; then move the corner, edge, or face to the desired location and execute. The corners, edges, and faces of an element such as a spaceform may be ambiguous in some orientations. An edge may be seen as a point, or a face may be seen as a line. In such cases rotate the element before shaping to overcome the ambiguity.



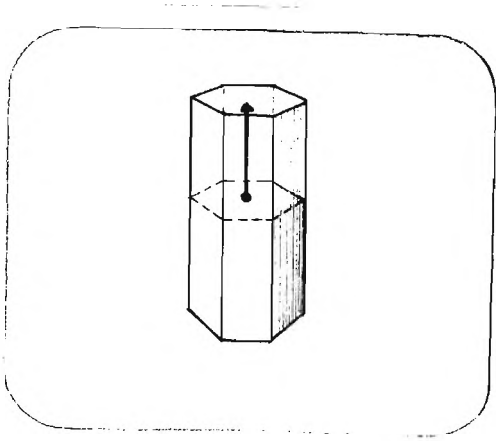
4.1

A cube can be contracted or expanded in any of six directions, two of which are shown in Drawings 4.1 and 4.2 To contract or expand, touch the face to be manipulated and "push" toward the cuboid for contraction, or "pull" away from the cuboid for expansion.



4.2

Rods and cylinders can be shortened or lengthened (along the axis of symmetry) in the same way by touching their end faces as shown in Drawing 4.3. They can be made thinner or thicker by similar manipulations. Touch the curved surface for cylinders or any side face for rods. For such manipulations rods and cylinders maintain a regular cross section. Spheres can be contracted or expanded by touching their surfaces or false edges and applying the same manipulations.



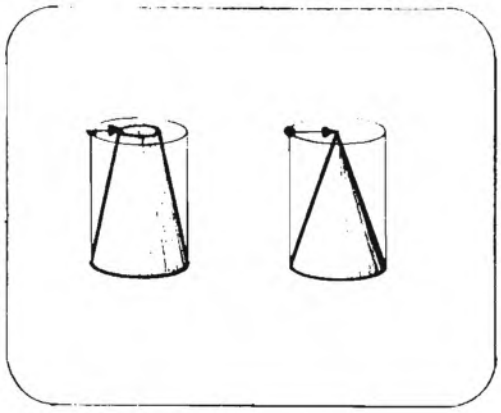
4.3

Further changes in the shapes of spaceforms can be made by working with the corners and edges rather than surfaces. Again, the designer must touch with the stylus the corner or edge to be affected and execute; then he moves the stylus to the desired position and executes again. It is assumed that during these manipulations the topology of the basic spaceform changes as the manipulations are carried out.

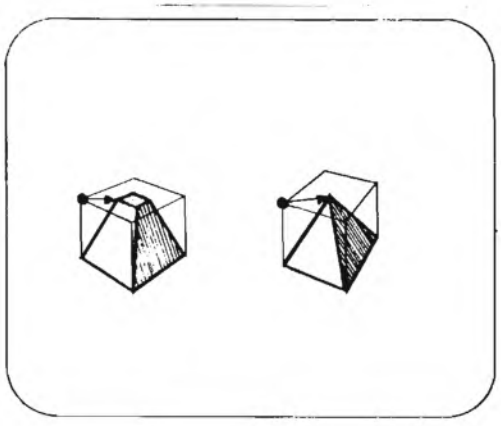
In the case of a cube, the designer may change an edge; but as he moves the edge, it remains parallel to its original position. This constraint limits the possibilities for shaping by contraction or expansion, but allows additional shaping to be accomplished by other manipulations. The cylinder has two real edges located where the top and bottom faces meet the curved surface of the cylinder. Manipulating these edges produces cones as shown in Drawing 4.4 on the following page.

By manipulating corners the designer changes basic spaceforms in still other ways. Moving a single corner of a cube toward the center of the surface brings all four edges together to form a pyramid. (See Draw-

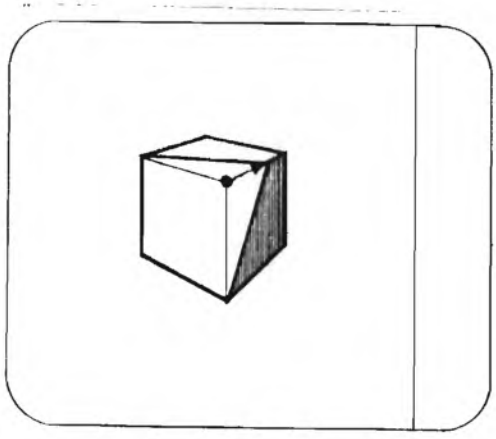
ing 4.5.) When the corner is manipulated along an edge, as in Drawing 4.6, a warped face is produced. By moving two consecutive corners the same distance along two edges, the designer shapes a wedge. (See Drawing 4.7 on the following page.) This wedge can also be created if only the top edge is moved in the manner described above. Rods can be altered in a similar manner as shown in Drawings 4.8 and 4.9. It is not possible to alter either the cylinder or sphere in this manner.



4.4



4.5

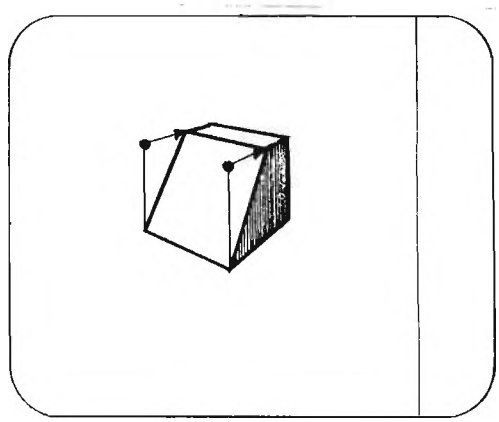


4.6

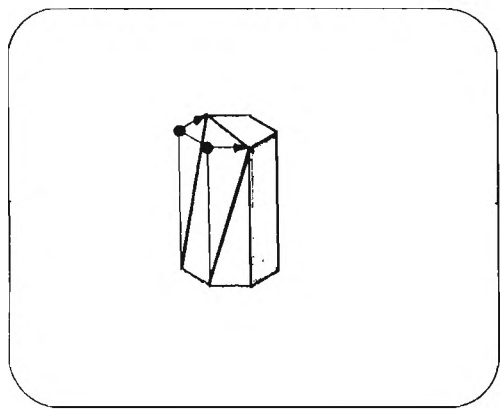
4.2 Size

It is also possible to shape spaceforms by assigning new dimensions to their edges. This is done by touching the edge with the stylus, executing, and typing LABEL SIZE. The length of the edge is then displayed as a label along the edge. To change the size, touch the label and type the

new dimension wanted. Both the element's size and its label change to reflect the new size. The direction of change is determined by the location of the stylus on the edge. For example, if the stylus is well to the left of the center of the edge, the change will be toward the left; if the stylus is clearly to the right, the change will be to the right. If the stylus is near the center, the change will occur equally on both sides (symmetrically about the center).



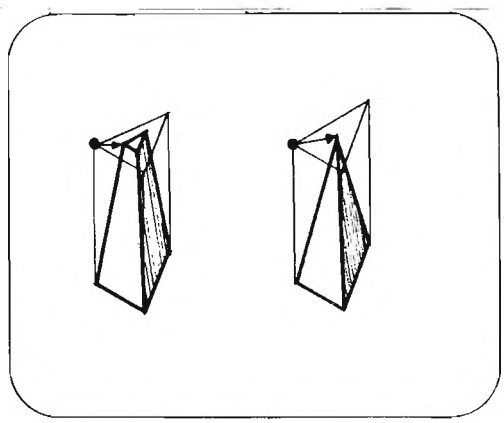
4.7



4.8

4.3 Cut

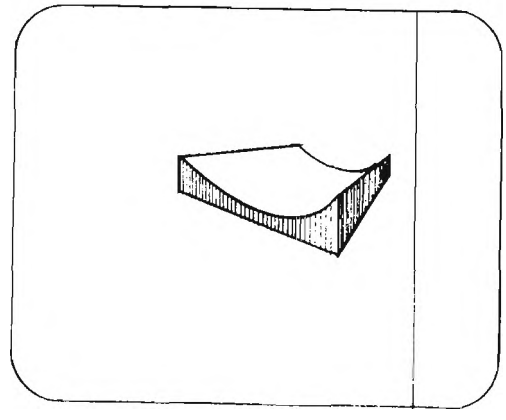
Any plane or curved surface may be a cutting plane used to slice through a graphic or building element. Cutting might be for shaping or for viewing. If it is for shaping, the unwanted portion is deleted; if for viewing, the obscuring portion is temporarily erased from the picture, but remains in the



4.9

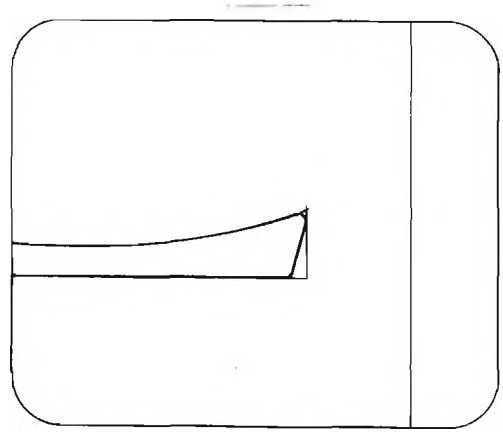
scheme. Cutting is done at right angles to the display plane. To cut, rotate the element into the desired position relative to the display plane. Retrieve a plane or curve from the key and place it. If the cut is for shaping, touch the unwanted portion, execute, and type the word ERASE. To restore an object cut for viewing to its original configuration, rotate the element to its position when cut, touch the cutting plane, execute, and type the word ERASE. This will erase the cutting plane and restore the element to its original configuration.

Drawing 4.10 shows a cuboid which has been cut with a curve to produce a cylindrical surface. The cuboid has then been rotated back into a perspective view.



4.10

The designer can create an infinite number of curved surfaces for such cutting by simply drawing the desired profile with the stylus. First, however, he rotates the object so that the direction of the cut is at right angles to the display plane—this is done by making some face or arbitrary surface coplanar with the display plane as in Drawing 4.11. Then he types DRAW, executes, and draws the curve he desires.



4.11

(He can smooth the curve by typing FIT and executing.) Now he can cut with the curve as described above. If the cut is not to extend the full depth of the graphic element, the element must be rotated so the direction of plane cut is parallel with the display plane, whereupon a dimension is applied to name the depth of the cut, and the element is rotated back for cutting.

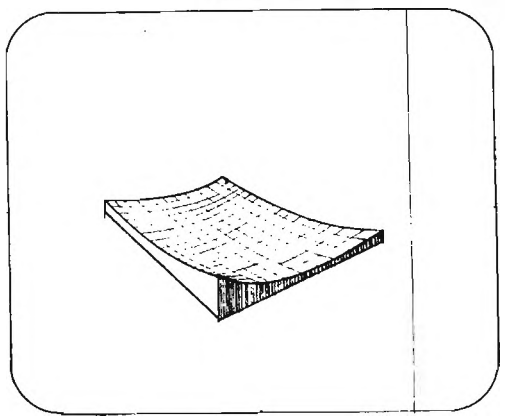
4.11

4.4 Draw

To draw is to describe a curve. To draw, type the word DRAW and then trace the desired curve with the stylus. The curve appears on the working surface as it is drawn. To fit the curve touch it, execute, type FIT, and execute.

4.5 Grid

Complex curved surfaces are more easily visualized and dealt with in terms of nets which are applied by gridding. Drawing 4.12 illustrates such a net on a curved surface.



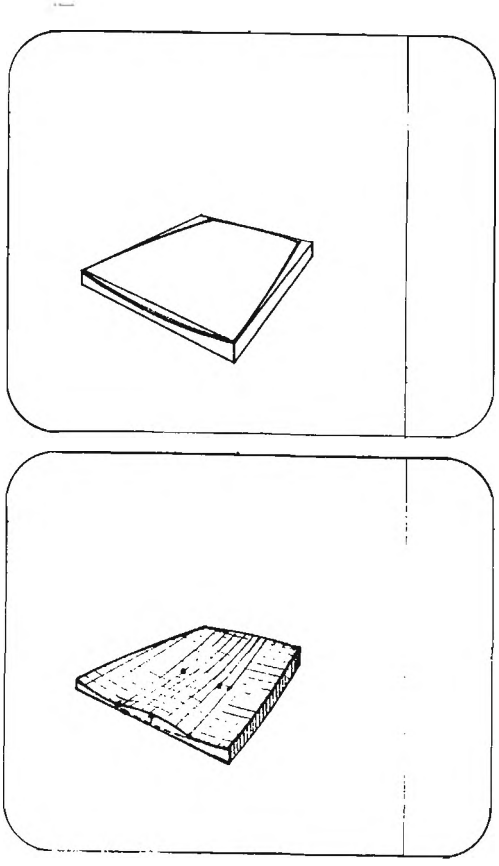
4.12

A typical net having three "columns" of spaces is available in the spaceform in-

dex for display in the key. To contract or expand a net on the working surface, touch one space and type the number of spaces desired. Repeat the process if the net is to be contracted or expanded in the other direction. To size the line segments, touch them, type both the word SIZE and the desired dimension. Give this command twice to size both sides of the rectangular space. When squares are wanted, the same dimension may be typed each time. Nets are always described as rectilinear (or square) in orthographic projection. In topography, for example, they are always rectangles or squares in plan.

4.6 Shape a Surface

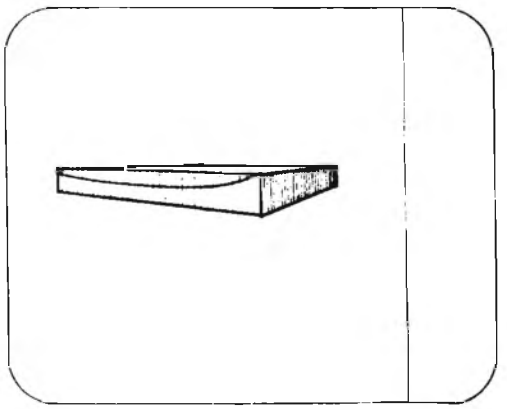
Occasionally the designer must deal with more complex curvilinear surfaces. Some furniture, for example, relies on full-scale mock-ups because drafting is inadequate. Topography is another example of complex curvilinear surfaces. Such surfaces can be developed from the basic or shaped spaceforms previously described by a combination of gridding, drawing, and other manipulations as in Drawing 4.13.



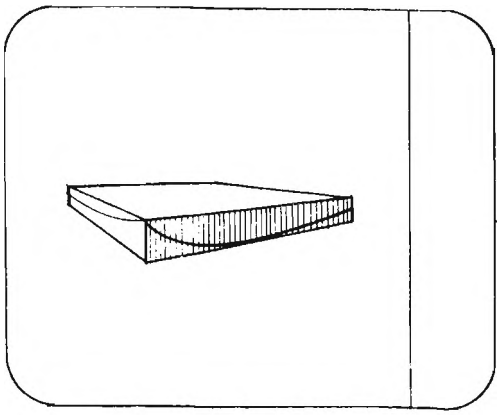
4.13

For such uses begin by positioning a cuboid on the

working surface and sizing it to desired dimensions. Make coplanar with the display plane one plane of the cuboid, then draw and fit the desired curve. (Drawing 4.14 shows such a cuboid after it has been rotated back into a perspective view.) Then rotate the cuboid until another face becomes coplanar with the display plane; draw and fit the second curve. (See Drawing 4.15.)



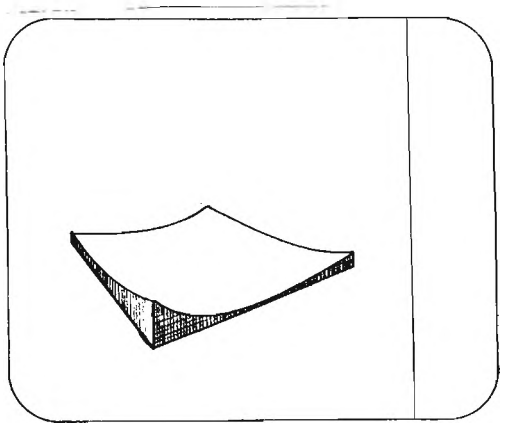
4.14



4.15

4.7 Retrieve a Surface

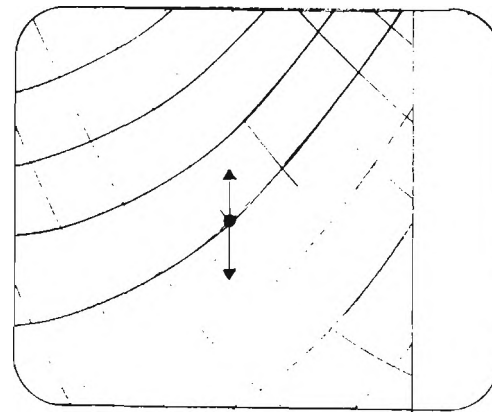
To retrieve a curved surface once all boundary curves have been drawn and fitted to a given spaceform, touch the surface, type RETRIEVE SURFACE, and execute. The spaceform shown in Drawing 4.16 results. The surface is defined by the four boundary curves and is that surface which would result if a perfectly elastic membrane were stretched across the curves.



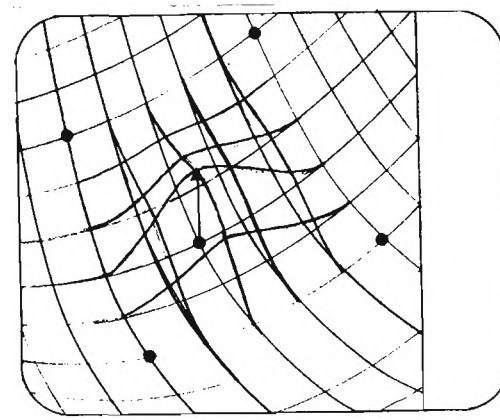
4.16

4.8 Reshape a Surface

If the surface formed by the perfectly elastic membrane is the surface needed, the designer may further shape the surface by attaching the stylus to any one of the nodes of the net and manipulating it up or down. Only vertical manipulation is allowed as shown in Drawing 4.17. The manipulation is as follows: touch each node at the periphery of the area which is to be reshaped and type FIX. Then touch in turn each node which is to be changed, execute, place, and execute again. The resulting curved surface is rounded on its crest and "faired" into the adjoining surface as illustrated in Drawing 4.18. If the designer wants to change the boundary curve, he may do so at any time by redrawing it. This of course affects the entire surface which must be retrieved anew. Several spaceforms, each with a shaped surface, can be packed to form a much larger surface which is faired together at the joints.



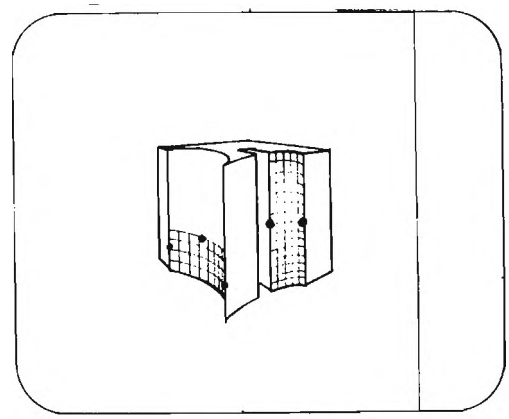
4.17



4.18

The designer is free to begin shaping, in a very general way, the object he is designing. When he has an acceptable general shape, he can then refine it to its most minute details. In the example shown, the

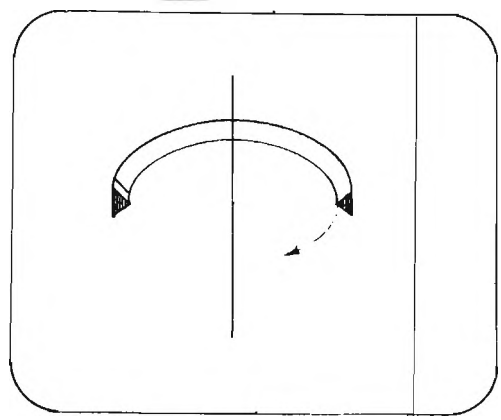
designer begins by shaping the general form of a chair seat by drawing the boundary curves. Now he may wish to give attention to the edges. To examine them he may have to enlarge the scale, and possibly, view various sections. These edges can be defined in the same manner as the first example given in Section 4.8. If the designer now wants to redefine the overall shape, he may do so by redrawing boundary curves and by retrieving new curved surfaces. This entire process is like carving the object out of homogeneous material with tools over which the designer exercises complete control. Even when a decision is made and a surface retrieved, the designer is not faced with an irrevocable situation. He can freely return to the previous state. Drawing 4.19 illustrates one of the many spaceforms which may be designed.



4.19

4.9 Revolve

To revolve is to develop a spaceform of revolution. To revolve, retrieve a line from the key and label it as an axis by typing AXIS. All revolution is done on the surface plane or on a plane parallel to it. Drawing 4.20 shows a typical spaceform of revoltuion. It appears



4.20

here in perspective, but it would be revolved in a top, or plan, view. Therefore, rotate the axis until it appears as a point. Retrieve the element wanted for revolution and place it relative to the axis, execute, and type the word REVOLVE.

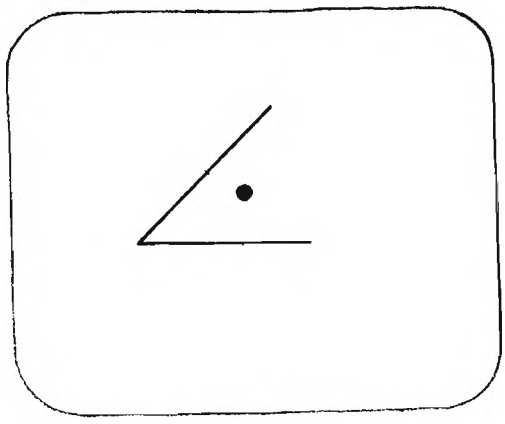
CHAPTER V

Assembling Manipulations

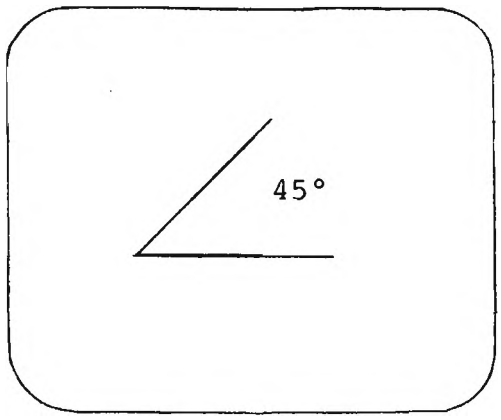
To assemble is to topologically combine primitive spaceforms or objects—these need not then be proximate.

5.1 Translate

To translate is to move an element in the display plane or in a plane parallel to the display plane. To translate, touch the element, execute, move the stylus to a new location, and execute again. The station and vanishing points of the translated element are those of the untranslated portions of the picture, which remain constant. Therefore, the convergence and foreshortening of the translated object constantly change during translation; but convergence and foreshortening of stationary elements remain unchanged. It should be noted that translating and placing are alike, except in translating the graphic element follows along with the movement of the stylus; whereas, for placing, the graphic element simply appears suddenly at the loca-



5.1

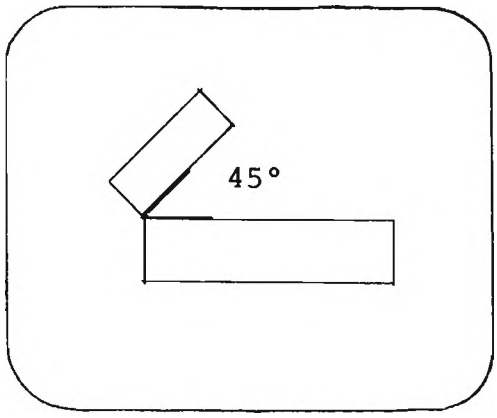


5.2

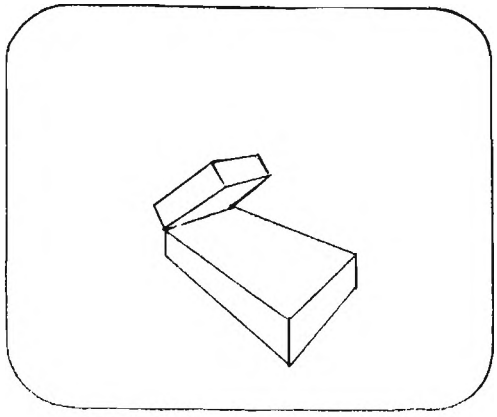
tion touched by the stylus.

5.2 Assemble at an Angle

To assemble at an angle is to put together two graphic elements at an angle. To do this touch the angle in the key surface and place it in the working surface as shown in Drawing 5.1. Assign the desired number of degrees, as in Drawing 5.2, touch that surface of one of the graphic elements to be joined, and then touch the leg of the angle to which it is to be coplanar. Next touch the surface of the second graphic element to be joined, and then the second leg of the angle (Drawing 5.3). Execute after each touch. The element thus assumes the desired relationship as illustrated in Drawing 5.4.



5.3



5.4

5.3 Make Coordinate

To make two points coordinate touch first one with the stylus, execute, and type COORDINATE; then touch the second one and execute. The first point moves to a position coordinate with the second. The

points may be the corners of spaceforms or objects.

5.4 Make Colinear

To make two lines colinear touch one line with the stylus, execute, and type COLINEAR; touch the second line and execute. The first line moves to a position colinear with the second. The lines may be edges of spaceforms or building parts.

5.5 Make Coplanar

To make two surfaces coplanar touch one surface with the stylus, execute, and type COPLANAR; touch the second surface and execute. The first surface moves to a position coplanar with the second. The surfaces may be faces of spaceforms or building parts.

5.6 Pack

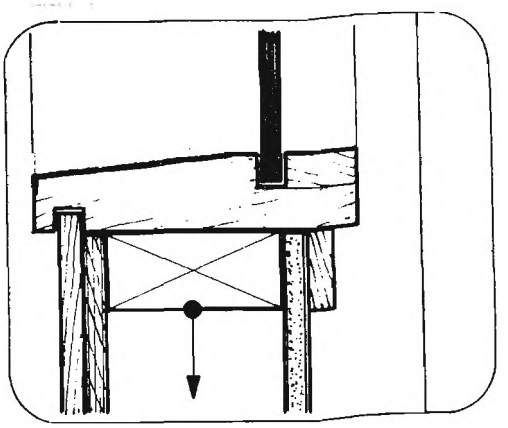
To pack is to position spaceforms or building parts into components without voids which are "unassigned"; for example, to pack rooms of a building touch and execute for each element to be packed, then type PACK. When the packing involves certain constraints such as minimal dimensions for certain elements, these constraints are input prior to packing.

5.7 Manipulate Clusters and Components

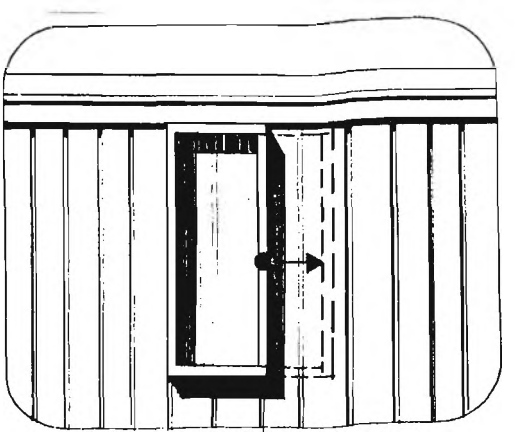
As previously stated, three portions of a graphic element can be touched unambiguously—the corner, edge, and face. These can of course be used whenever a corner, edge, or face is to be manipulated. When an entire element or group of elements is to be manipulated, however, the element or group must first be identified by naming. In the abstract

mode groups of basic and shaped spaceforms are called clusters; in the concrete mode groups of parts (objects) are called components. Once named, clusters and components are treated as topologically combined elements. Clusters have only the geometrical attributes of extent and shape, but components take on any attributes available in the object system process data.

To reduce the ambiguities in naming it is helpful to work at an appropriate level of detail. A component such as a window, for example, has a topologically fixed arrangement of sill, jambs, head, muntins, and glazing, which must constantly be maintained even when the size or location of the window is changed. Generally it is best to make changes relative to a view at the proper level of detail. If the two-by-four supporting the sill is to be lowered, use a cross-sectional detail as in Drawing 5.5; if the jamb is to be moved, use a larger detail showing the window as a whole (Drawing 5.6); and, if the entire window is to be translated, use a still larger detail showing the window in relationship to the wall as in Drawing

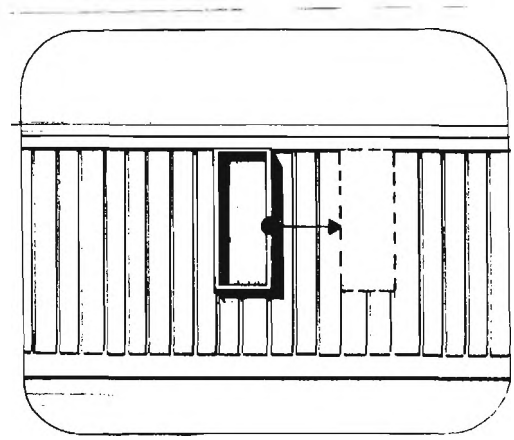


5.5



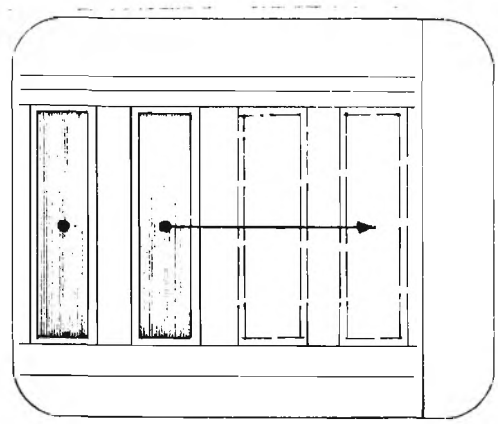
5.6

5.7. In Drawing 5.8 two windows are pointed so they form a "component" which is then translated to the right. Even at the appropriate level of detail, however, it is necessary to name the element to be manipulated——for example, the TWO-BY-FOUR, JAMB, or WINDOW.

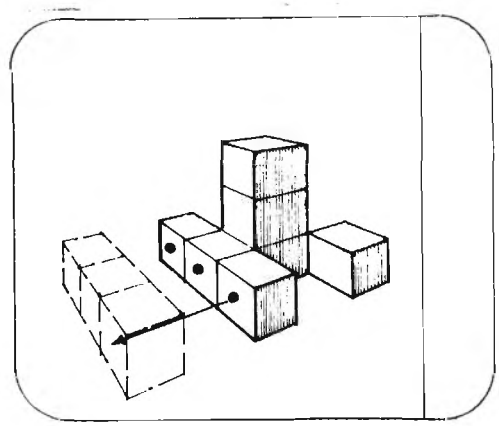


5.7

Clusters have no established names as do components, but they may be assigned arbitrary cluster names such as CLUSTER 1, CLUSTER B, and the like. To name clusters, point and execute in turn all the spaceforms to be included in the cluster, then name them in the prescribed manner. The illustration at the right (Drawing 5.9) shows a cluster of three cuboids and the way this cluster may be translated.



5.8



5.9

CHAPTER VI

Viewing Manipulations

Since design ultimately involves the construction of objects in the real world, every attempt should be made to represent real world conditions at the display. Central to this idea is the necessity for viewing the object or building under design three-dimensionally. A single view is not sufficient for complete visualization. The designer should be able to rotate his scheme about three axes for any other desired view and to zoom toward or away from it. Scrolling, enlarging-reducing, rotating, zooming, scanning, and panning are discussed in this section of the paper. These viewing manipulations (and the orienting manipulations discussed in Chapter 7) differ from the shaping and assembling manipulations in that the former do not affect the description of the building in the project file, but the latter do.

6.1 Scroll

To scroll is to bring portions of a picture not previously visible into view on the display, as in the case of text which can be scrolled into view continuously. This is accomplished by pressing the "scroll" button located at the track ball and then moving the track ball. Only orthographic projections, text, and numbers can be scrolled. Only up-down and side-to-side scrolling is possible.

6.2 Enlarge; Reduce

To enlarge is to increase the size of a displayed picture. To enlarge a picture press the "enlarge-reduce" button and move the foot ped-

al with the toe. For this process the station point and vanishing points may be thought of as part of the object because an enlargement of the picture causes an equivalent enlargement of the imaginary "triangle" bounded by the station and vanishing points.

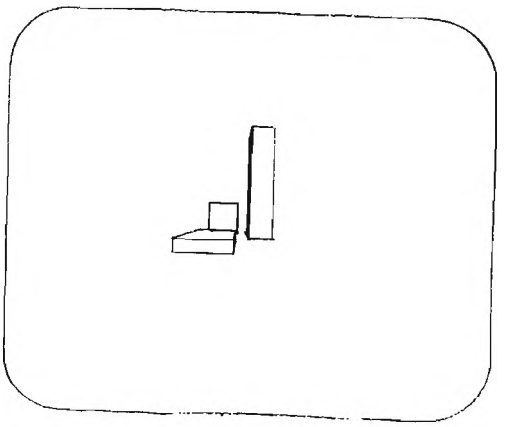
To reduce is to decrease the size of a displayed picture. To do this, press the "enlarge-reduce" button and move the foot pedal with the heel. For this process also, the station point and vanishing points may be thought of as part of the object because a reduction of the picture causes an equivalent reduction of the imaginary triangle bounded by the station and vanishing points.

6.3 Rotate

To rotate a part or all of an element (object) press the "rotate" button located near the track ball and rotate the ball in the desired direction. The element will rotate about a point near its centroid. If the designer wishes to rotate about any other point, he can retrieve a point from the key, label it CENTROID, place it, execute, press the "rotate" button, and then rotate the track ball. To rotate a portion of an object, touch the stylus on the desired part or parts, execute, and rotate. The centroid is the center of visual mass of an element. Drawing 6.2 shows a rotation of the cuboids of Drawing 6.1.

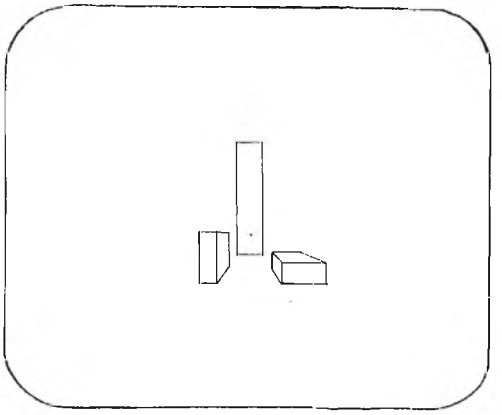
6.4 Zoom

The designer may wish to examine his scheme more closely, or he may wish to move away to include more of the scheme and its surroundings on the display surface. By zooming in or out he can move toward or away from the object being designed.



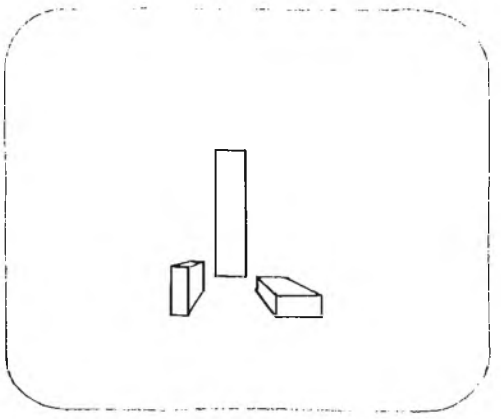
6.1

Zooming is always directed toward the center of the working surface. If a zoom in a different direction is wanted, the designer first scans until the desired zoom point is centered in the working surface. In this manner, he is able to select specific details or parts and enlarge them until they alone occupy the display.



6.2

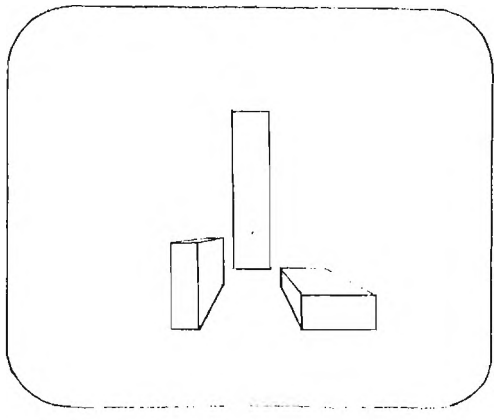
Zooming appears as shown in Drawings 6.2 through 6.5. In these illustrations the designer zooms toward three basic spaceforms as shown in the consecutive views.



6.3

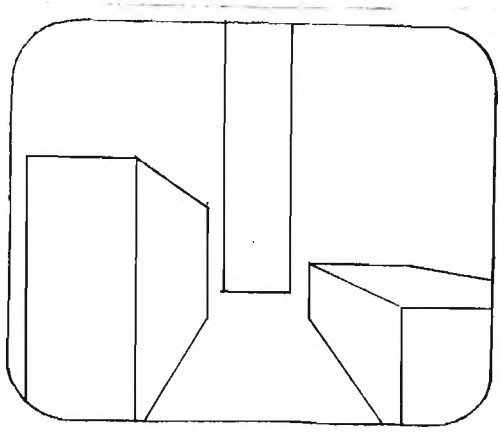
Zooming causes two additional effects which must be

dealt with: lines which should no longer be visible on the display surface as the object is approached; and increased detail as it is seen more closely.



6.4

The first effect requires clipping to remove those lines which should no longer be visible. In zooming toward a building facade, for example, the designer might begin with the entire facade included in the picture and then wish to zoom toward a door. As he "approaches" the door, it occupies more and more of the working surface; so the parts of the elevation peripheral to the door must be clipped. At some time only the door and frame ought to be displayed, and at a later time just the door itself. As the designer zooms back away from the door, the effect is reversed, and clipped lines must be replaced in the picture so that it continually floods the CRT.



6.5

The second effect is that of increasing detail while zooming in. When the entire elevation is in view the doorknob might well be erased from the picture; but when zooming in the doorknob should appear as added detail. The accommodation of this effect serves two purposes:

first, it tends to represent a real world visual effect; and, second, it minimizes the number of lines displayed at any time.

6.5 Pan

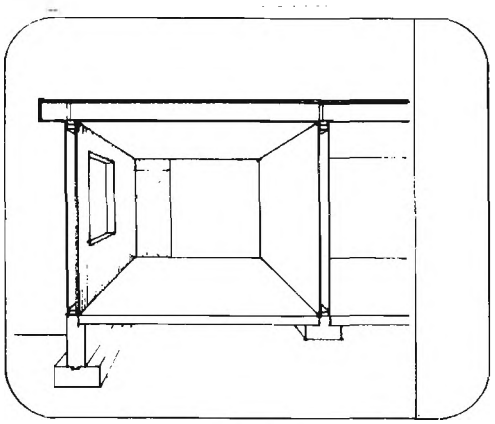
The designer often wants to view a building he is designing from a single vantage point. Panning permits him to direct his attention to various building parts without changing his location relative to the building. Panning represents a horizontal rotation of the head of the viewer. To pan, press the "pan" button located near the track ball and rotate the track ball. In panning, the relative positions of the station and vanishing points remain constant, but they move in a circular arc with the station point as the axis of revolution. Objects at the same distance from the viewer retain the same relative size.

6.6 Scan

When viewing a building at an enlarged scale, it is not possible to get the entire picture on the working surface. Scanning allows the designer to see another area of interest without changing the viewing distance. Each view is in perspective and represents one's walking or driving past the building. To scan, press the "scan" button near the track ball and move the track ball in the direction of simulated movement. In this process, vanishing points retain a fixed relative position, and the station and vanishing points "move with the viewer." Graphic elements retain the same apparent size during the scanning, and the translation of the graphic elements is opposite to the simulated movement of the viewer.

6.7 Cut to View

For greater visualization of schemes it is necessary to cut sections quickly and easily. The capacity to do this allows views of the scheme which would otherwise be hidden. For a description of cutting to view refer to Section 4.3 Cut. (See Drawing 6.6.)



6.6

6.8 Sectional Zoom

Sectional zooming is a combination of cutting to view and zooming. This process permits a representation of movement into a space through the faces which enclose it. Manipulation is done by means of the foot pedal as with conventional zooming, but the command SECTION must be typed before zooming.

CHAPTER VII

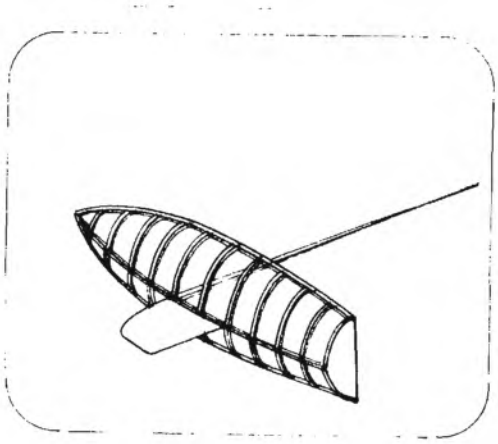
Orienting Manipulations

7.1 Introduction

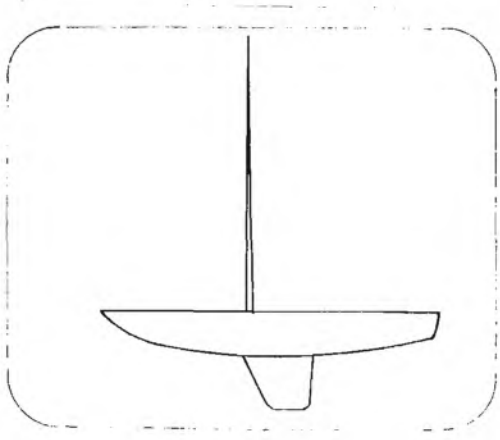
To negotiate the real world we rely upon all our senses—vision, smell, taste, hearing, touch, and the kinesthetic sense—and a full range of environmental cues. These enable us to perceive and deal with the world. Computer graphics, however, calls upon our visual sense alone; and even at that a computer graphics picture lacks many of the visual attributes of a real world situation. Some of the visual attributes which make real world objects recognizable are: foreshortening and convergence of perspective; grain and texture; stereoscopic quality of two-eyed vision; variation of views as body, head, and eye move; hue, value, and chroma; atmospheric graying of objects distant from the eye; shade and shadow; overlapping of nearer objects over more distant ones; etc. Furthermore there are characteristics of computer graphics, such as the hidden lines [10], which do not occur in hard copy graphics.

Because of the relative speed with which he can build up and manipulate spaceforms (which appear dynamically as perspectives rather than as static orthogonals), the designer can easily become disoriented with respect to his pictures. The spaceforms appear to be floating in outer space, and the designer must have ways to establish spatial cues in order to relate one picture part to another, or the whole picture to a reference plane such as the horizon plane. This is especially true when (1) edge drawings rather than halftones are used; (2) hidden lines are not removed; (3) no ground plane to serve as a reference plane is

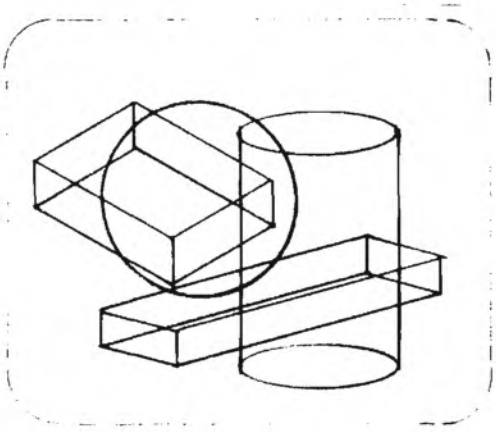
given; or (4) the graphic elements are abstract rather than concrete and therefore have no meaningful, readily discernible characteristics. For example, a picture of a sailboat might have a pointed bow and a flat stern which orient the designer to front and rear no matter what the view of the boat. (See Drawings 7.1 and 7.2.) But this would not hold for boxes, cylinders, or spheres in an abstract cluster as in Drawings 7.3 and 7.4. Drawing 7.3 illustrates depth ambiguities which occur when hidden lines are not removed.



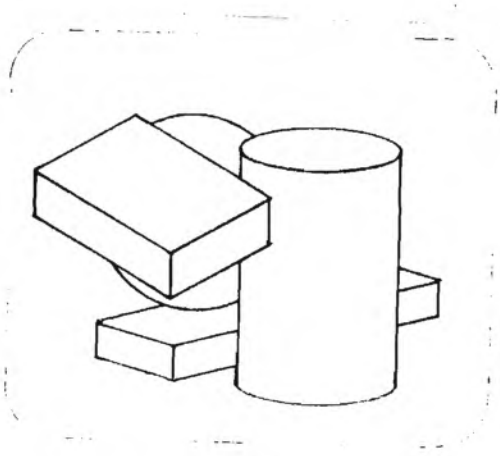
7.1



7.2

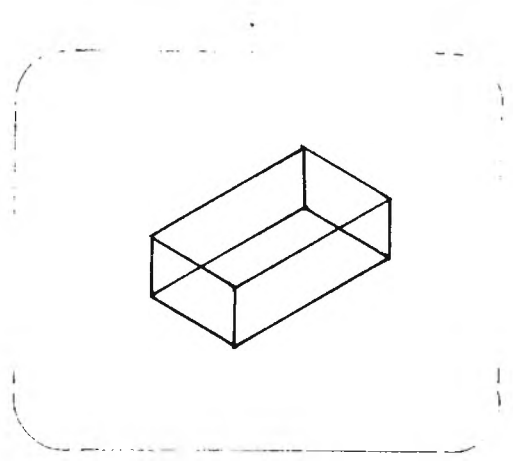


7.3



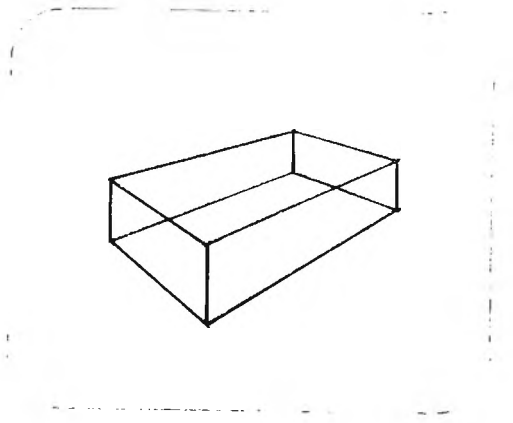
7.4

When certain visual cues are missing, pictures are ambiguous; i.e., they might be interpreted several ways. The simplest example is the Necker Cube [11] as shown isometrically in Drawing 7.5. It is impossible to determine which are the "hidden lines." Rather we see the cube at one moment from above and a moment later from below; and the two percepts alternate constantly as we stare at the cube.



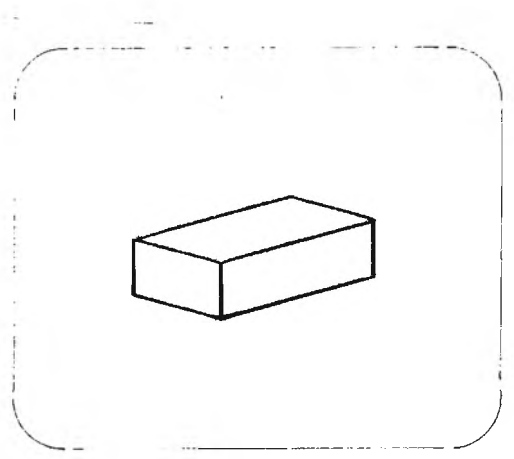
7.5

In Drawing 7.6 the same cuboid is shown in perspective. Here reversal occurs also. The small face appears first as the far end of a box in perspective and later as the near end of a truncated (and somewhat skewed) pyramid.



7.6

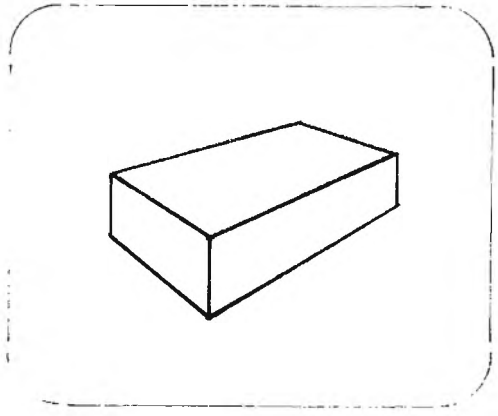
Drawing 7.7 shows the cube in an isometric view with hidden lines removed. The picture is still ambiguous when conceptualized as consisting of three volumeless faces (rather than six as needed for a closed box) that may be seen from above or below. This reversal



7.7

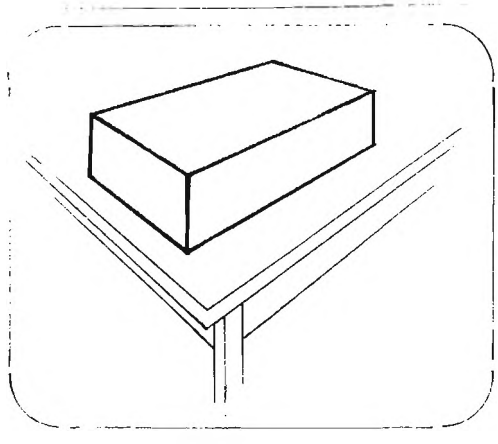
makes the faces appear first as convex and then concave.

Drawing 7.8 shows the cube in perspective with hidden lines removed. Here it is somewhat more difficult to think of the cube as consisting of three faces viewed from below. To do so we would have to imagine the two sides to be tapered rather than rectilinear.



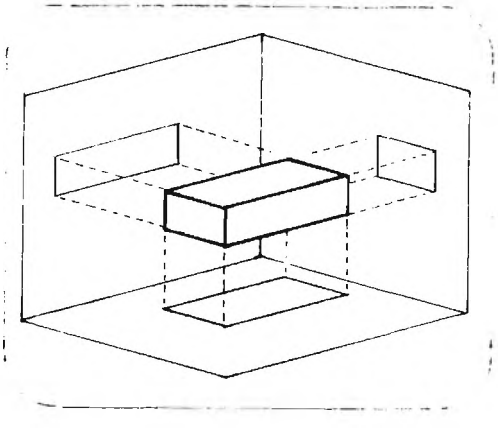
7.8

In Drawings 7.5 through 7.8, no reference plane appears. In Drawing 7.9, the cuboid is shown with a table top, and the most likely interpretation is that it rests directly on the table top. Alternatively, it may be visualized as a small object floating above the table top.



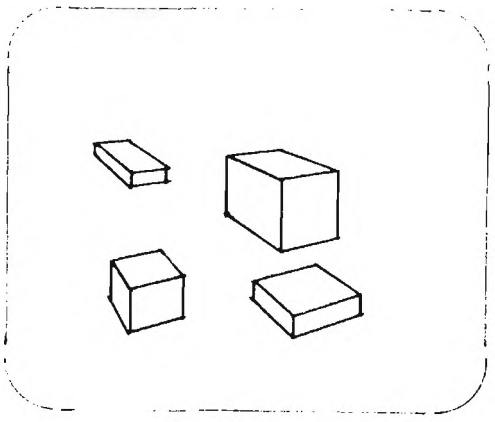
7.9

Drawing 7.10 shows the cuboid in relationship to X,Y, and Z coordinate planes. Orthogonal projections of the object on these planes describe the object itself and assist the viewer in seeing the cube with respect to the coordinate planes. Such planes need not be considered as additions to SPACEFORM, but rather as three volumeless cuboids which can be constructed when wanted by the designer.



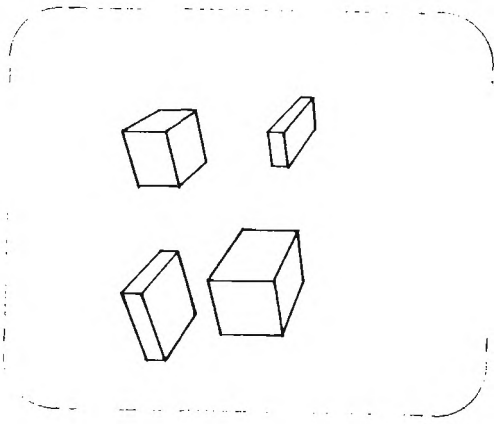
7.10

The need for reference planes is more dramatically demonstrated in Drawings 7.11 through 7.14. Four cuboids have been positioned in Drawing 7.11. These cuboids have fixed relative locations in the project file, but the designer may lose orientation with respect to the cuboids if they are rotated, as in Drawing 7.12 where they have been rotated ninety degrees.

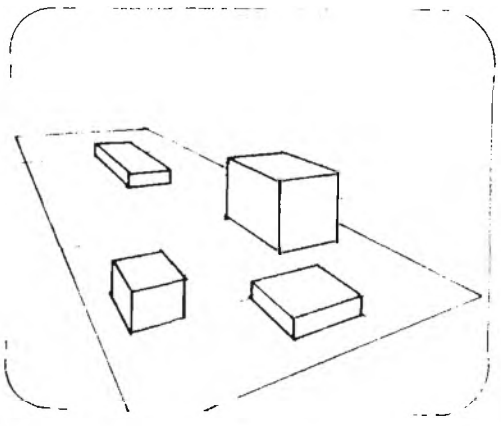


7.11

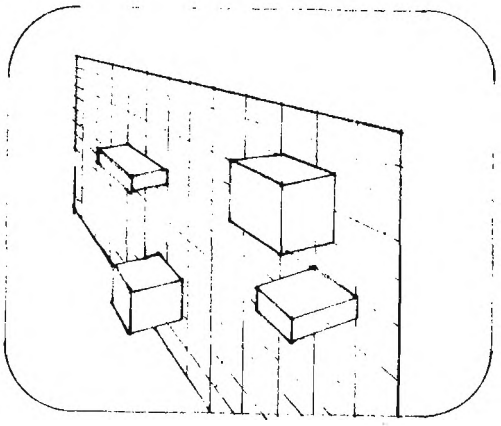
To dramatize the ambiguity Drawing 7.13 includes a horizontal plane to which each of the four cuboids of Drawing 7.11 is attached; alternatively, Drawing 7.14 shows a vertical plane to which they are attached. The two, of course, are completely different interpretations of Draw-



7.12



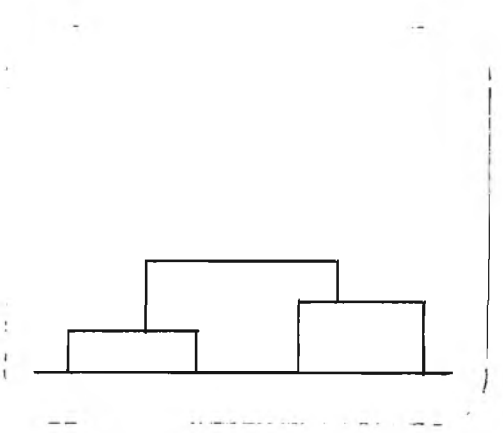
7.13



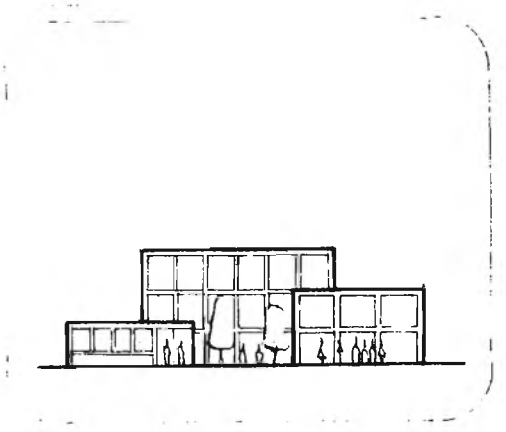
7.14

ing 7.11; but the presence of a reference plane eliminates, or at least reduces, the ambiguity.

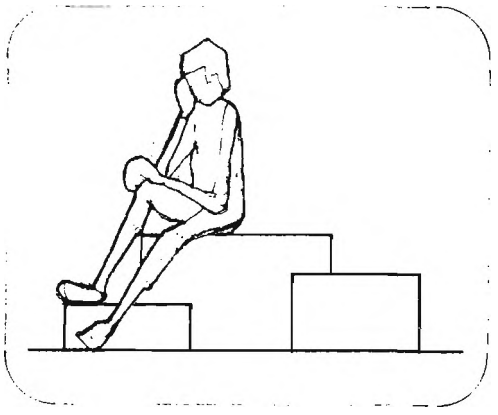
Scale establishes size relationships between humans and objects. Drawing 7.15 shows three cuboids in facade view. The scale is unknown. Cues to scale can be established by the addition of details such as doors and windows in Drawing 7.16 or by introducing human figures as in Drawing 7.17 and 7.18.



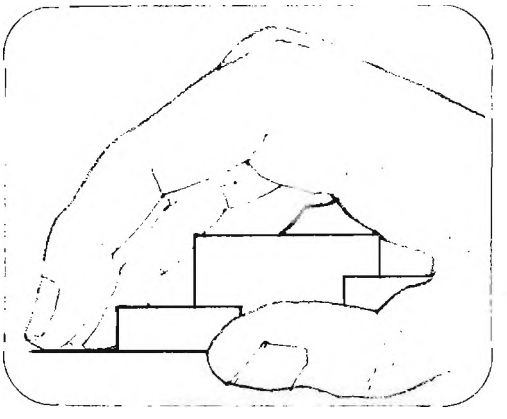
7.15



7.16

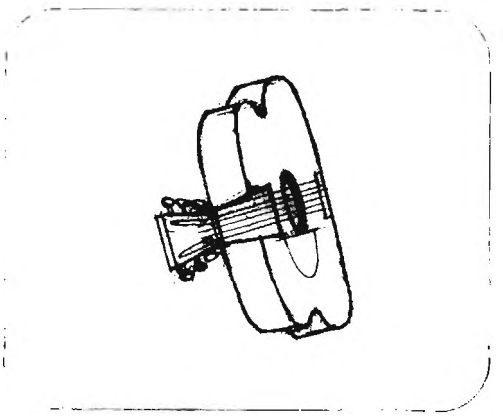


7.17

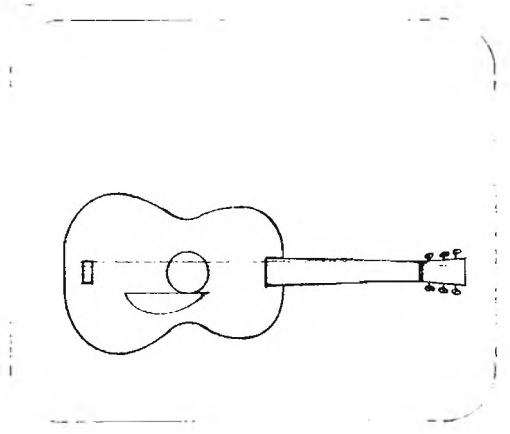


7.18

Orthogonal projections are used almost exclusively in hard copy drawings for object descriptions. The SPACEFORM graphics language of ARCAID offers a greatly increased use of perspectives instead. This does not mean orthogonal projections will be eliminated, however, for they still may be needed for clear interpretations. To manufacture the top of a guitar, for example, the orthogonal projection of Drawing 7.20 is obviously superior to the perspective of Drawing 7.19.



7.19



7.20

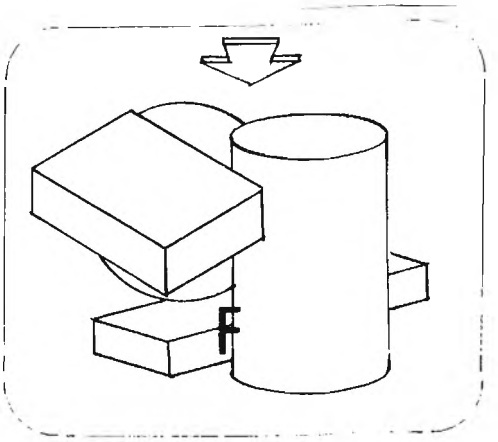
As previously described, there is a display plane coplanar with the display surface, and this can also be used for orientation. For instance, it can be used for securing orthogonal projections. To do this, simply touch the surface or face for which an orthogonal is wanted, and then type MAKE COPLANAR WITH DISPLAY PLANE. The surface or face touched assumes the desired location.

Still other orientations are those to the front, side, or back of an object or graphic element, or to compass points. Again, no special elements nor manipulations are needed. The "front" of a graphic element can be established at its real front or at any arbitrary face by attaching either a spaceform shaped as an "F", or any other distinctive spaceform.

Compass points can be established with an arrow assembled from a cuboid and a triangular rod. The arrow may be attached to a cluster, component, or scheme; or it may be positioned relative to a reference plane, coordinate planes, or a site.

See Drawing 7.21.

Scale may be handled similarly. In the early developmental stages of ARCAID a crude human figure can be constructed from cylinders and spheres. Later a more precise animated figure can be added to the ARCAID library.



7.21

7.2 Orienting

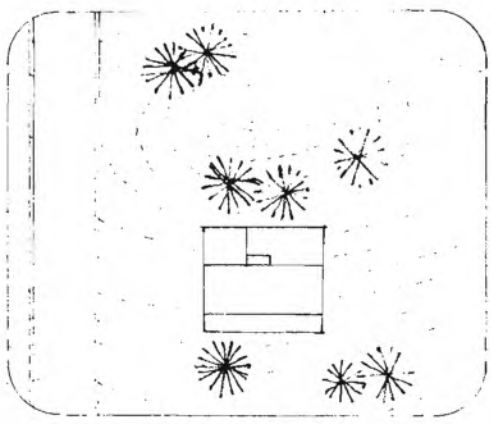
The preceding discussion has treated questions of ambiguity in computer graphic pictures. At any time during design, the building scheme or graphic element has topological relationships which have been established by shaping or assembling manipulations, and often the ambiguity is based on a particular method of viewing. In most cases further viewing removes the ambiguities and reveals the true form of the building or graphic element.

In certain cases, however, it is necessary or desirable to provide a site or other reference plane or a set of X, Y, and Z coordinate planes. In such cases no special manipulations are needed; previously described ones will serve.

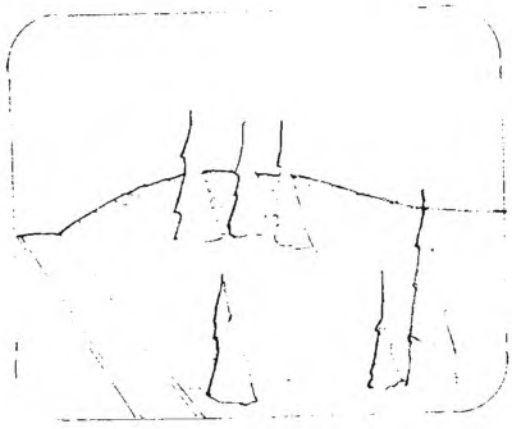
To establish coordinate planes, simply touch and assemble three cuboids with null thickness such that the adjacent cuboids are perpendicular. These can be named as a cluster independent of the objects.

A single reference plane is created in the same way.

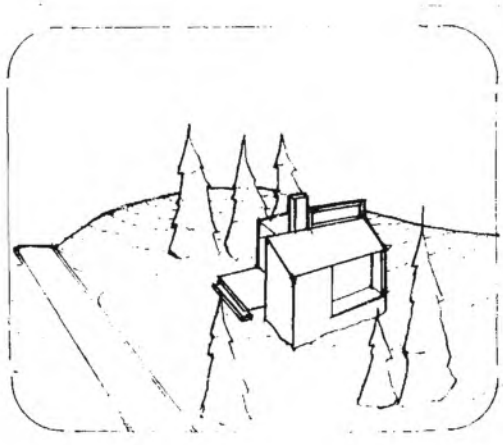
Drawings 7.22 through 7.24 show how a site serves as a reference plane and how rotation of the picture reduces ambiguities. Drawing 7.22 shows a house on a site for which contours are shown dotted. Since contour elevations are not given, it is impossible to tell whether the concentric ovals at the top of the picture represent a mound or a [12] depression. Drawing 7.23 shows the site in perspective from a frontal rather than a top station point, and clearly reveals that the contours represent a hill. In Drawing 7.24, the house appears. It is obvious that such viewing capabilities are extremely helpful as a design tool. The facility to rotate repeatedly from orthogonal to perspective views permits those continuous scheme changes which are necessary for perfecting the scheme and wedding building to site.



7.22



7.23



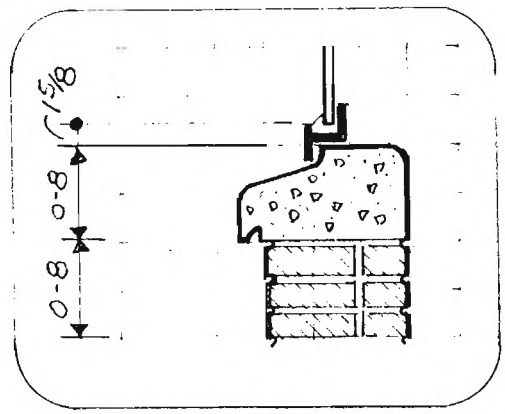
7.24

CHAPTER VIII

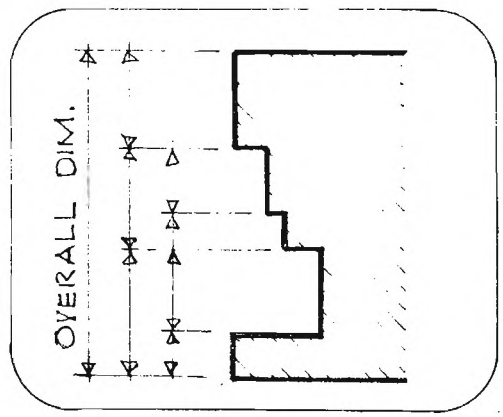
Miscellaneous Manipulations

8.1 Dimension

The contract-expand and size manipulations establish the extent of the spaceforms and objects; but the dimension manipulation assigns arrows, numbers, and, when needed, words to "name" these extents. For arrows the [13] modular convention is adopted in which dimensions of module lines terminate in triangles, and dimensions of points between module lines terminate in dots. (See Drawing 8.1). For numbers, the conventional form 8 - 2 replaces the traditional form 8' - 2" to indicate eight feet and two inches. The dimension manipulation is only available for elements in orthographic projection; this is also true for labeling of dimensions. Dimension arrows for a string of elements are always aligned; those for details appear near the particular spaceform or object; and those arrows for overall dimensions appear away from the spaceform or object, as in Drawing 8.2.



8.1



8.2

To perform the dimension

manipulation, touch the line (or the point) where the dimension arrow is to begin, type DIM, and execute. The dimension arrow will appear as a rubber band line. Next touch the line (or the point) where the dimension arrow is to terminate. Then type LABEL and the length of the arrow will appear. If the length is not as desired, the element must be re-sized.

8.2 Plot

The plot command permits the plotting of hard copy drawings of schemes or parts of schemes which have been developed at the CRT. The plot command transfers control from the CRT compiler-processor to the plotter compiler-processor; it transmits parameters which select the desired views and positions the parameters on the plotter paper. These parameters include border, title block, starting point for the lower left portion of the drawing, type of drawing (whether orthogonal or perspective), names, notes, dimensions, and the like.

CHAPTER IX

Elements in the SPACEFORM Language and ARCAID

To some degree elements have already been discussed in connection with the various manipulations. For this reason the present chapter is devoted to definitions with a minimum of supplementary text.

9.1 Element: the generic term for any textual, numeric, or graphic element including any component or system.

9.2 Letter: any alphabetical character.

9.3 Number: a digit; or any combination of digits, commas, and decimal points.

9.4 Primitive: a point, line, curve, net, angle, or surface. Primitives are available in the SPACEFORM Language index.

9.4.1 Point: a graphic element having only a location in space.

Corner: a point which defines the intersection of two or more edges of a spaceform or object.

9.4.2 Line: a graphic element defined by its endpoints; a straight line segment.

Edge: a line which defines the intersection of two faces of a spaceform or object, or the bounds of a surface.

False edge: a line, not an edge, which distinguishes a surface from its ground. In most perspective views of a cylindrical spaceform, for example, one end appears as a distorted ellipse and the opposite end as a segment of a

distorted ellipse. These are joined by two straight lines which delimit the cylindrical surface from the ground. The straight lines are false edges. As the cylindrical space-form is rotated on its axis of symmetry, the false edges retain their relationship of figure to ground while constantly changing their relationship to the curved surface.

Axis: a line which has been designated as an axis by a name.

9.4.3 Curve: a graphic element defined by its end points and equation; a curved line segment. The two types of curves are boundary and primitive curves.

Boundary curve: a curve which defines the extent of a curved surface.

Primitive curve: a circle, ellipse, parabola, hyperbola, or catenary. Primitive curves are available in the graphics index.

Drawn curve: a curve which results from freehand drawing movements of the stylus as constrasted with the primitive curve, which is selected from a set of curves and adjusted as needed.

9.4.4 Surface, Plane

Surface: a plane or a curved surface.

Plane: a graphic element that wholly contains every straight line joining any two points that lie within it.

Curved surface: an infinite aggregate of points constituting a space of two dimensions.

Boundary defined surface: a surface developed by defining boundary curves. The resulting surface has the shape of a perfectly elastic membrane stretched between the boundary curves.

Face: one of the surfaces which bounds a spaceform or object.

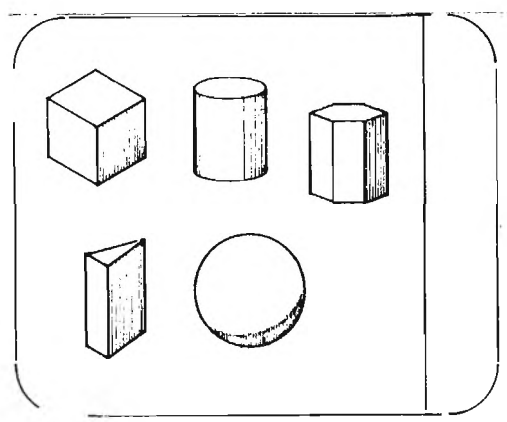
9.4.5 Angle: a primitive consisting of two intersecting lines. The angle degrees can be assigned by touching between the lines and typing the number of degrees desired. The lines assume the typed angle and retain it until a new number of degrees is assigned.

9.4.6 Surface net: a two-dimensional grid of rectangles which may be applied to any surface. A surface net permits reshaping of the surface by moving net nodes.

Net: two families of parallel lines intersecting at right angles and lying in a plane.

Net node: an intersection of two lines of a surface net. By touching nodes with a stylus and pulling up or down the surface can be reshaped.

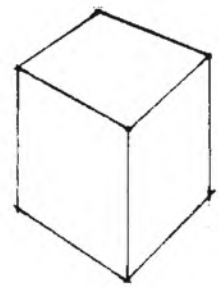
9.5 Basic spaceform: any topologically described, simple, geometric solid such as a cube, cylinder, sphere, hexagonal rod, or triangular rod. Basic spaceforms have unit dimensions and occupy an area about one-inch square in the key. (See Drawing 9.1)



9.1

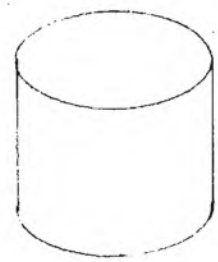
9.5.1 Basic spaceforms

Cuboid: a spaceform defined by six faces, all of which are rectangles. That cuboid which is a basic spaceform, has unit dimensions and occupies an area of approximately one square inch on the key surface. Shaped cuboids can assume any desired values for any three dimensions. A cuboid having one null-valued dimension is a rectangle; a cuboid having two null-valued dimensions is a line. [14]



9.2

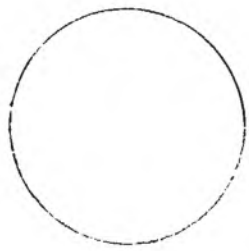
Cylinder: a spaceform defined by two faces which are circles and one face which is a right cylinder. That cylinder which is a basic spaceform, has a unit diameter and altitude and occupies



9.3

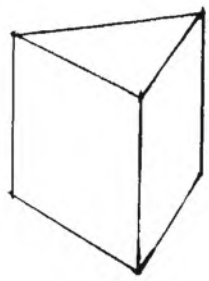
an area of approximately one square inch on the key surface.

Sphere: a spaceform defined by a sphere. That sphere which is a basic spaceform has a unit diameter and occupies an area of approximately one square inch on the key surface.



9.4

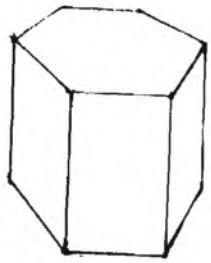
Triangular rod: a spaceform defined by two faces which are equilateral triangles, and three which are rectangles. That triangular rod which is a basic spaceform has unit dimensions and occupies an area of approximately one square inch on the key surface.



9.5

Hexagonal rod: a spaceform defined by two faces which are regular hexagons and six which are rectangles.

Sides of the hexagons have a half-unit dimension, and the rod has a unit altitude. It occupies about one square inch on the key surface.

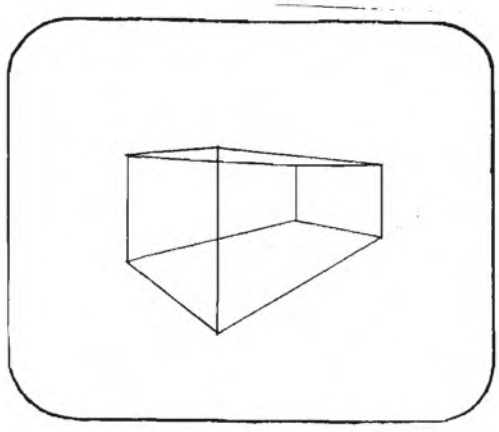


9.6

9.5.2 Types

There are three types of spaceform--frame, box, and solid. Both basic and and shaped spaceforms having any shape whatever are available in these three types. Objects, however, are generally built up of solids for which a particular material can be assigned.

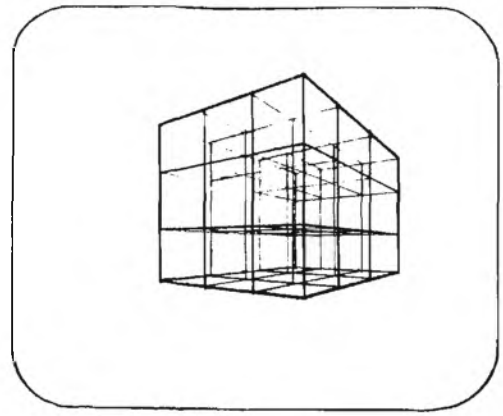
Frame: a spaceform which is defined by its edges only. Edges at the rear of the frame are visible unless concealed by forward edges as shown in Drawing 9.7.



9.7

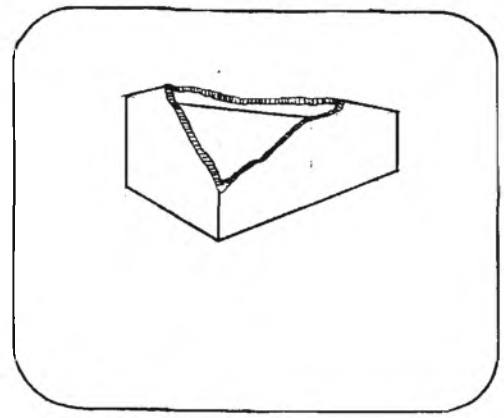
Frames are particularly useful for the design of trusses, rigid frames,

and other structural systems as illustrated in Drawing 9.8. At the outset, the edges of the frame may be thought of as cuboids having null dimensions in cross-section. As the design proceeds, these dimensions can be assigned, automatically "fattening" the members represented by the edges.



9.8

Box: a spaceform which is defined by opaque faces. A hole cut in a face of a box reveals the interior as in Drawing 9.9.

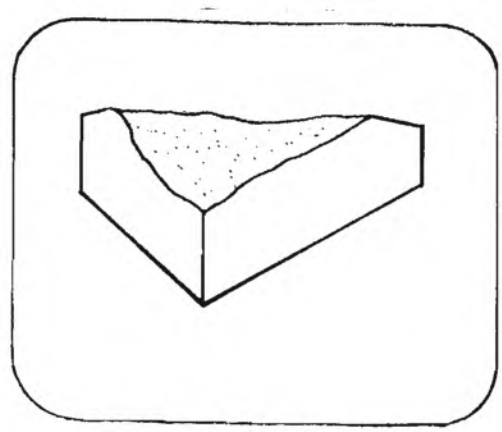


9.9

Boxes are convenient for representing the enclosures of rooms or buildings, since a hole cut in any face may represent a door or window which exposes to view the interior of the enclosed space. The faces of the cuboid box may be thought of as cuboids having a null thickness, which thickness can

later be assigned as appropriate for floors, walls, and ceilings, automatically "fattening" the parts represented by the faces.

Solid: a spaceform of solid matter. A hole cut in a solid reveals a pattern of points and lines which represents solid matter as in Drawing 9.10. It should be noted that this definition conforms to usage of the word solid in conjunction with physical matter rather than with geometry. Geometric solids are hollow and are referred to herein as boxes.



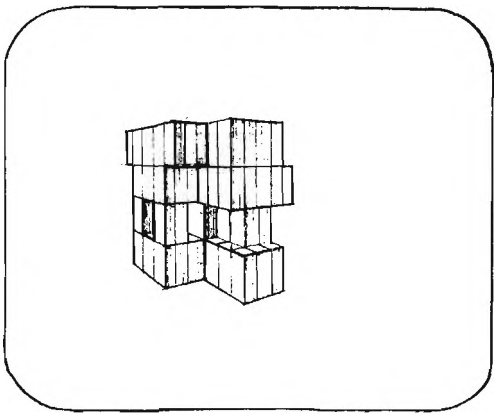
9.10

Solids are most useful spaceforms for these parts of the scheme which are to be assigned some material. The edges of a frame, or faces of a box may later be assigned such a material and, thereby, converted to solids.

The spaceforms of Drawing 9.11 could be either boxes or solids. If the cuboids represent rooms, boxes are better; but if they represent granite blocks in a wall, solids are more appropriate.

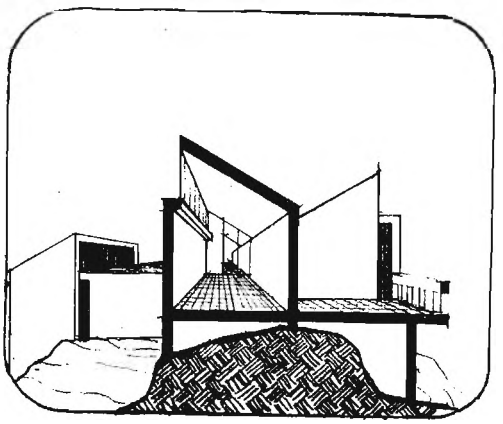
9.6 Shaped spaceform: any basic spaceform which has been modified; or any complex spaceform built up from primitives and which is not an object.

9.6.1 Spaceform of revolution: a shaped spaceform which has been created by revolving a graphic element about an axis.



9.11

9.7 Object: anything named as an object or topologically described with attributes extending beyond the geometric. Objects may be displayed on the key and working surfaces. Spaceforms are abstract, having only shape and size; but objects have additional attributes and, therefore, are more concrete. At the display we see images of objects, but we name them and otherwise deal with them as if they were the objects themselves. Draw-



9.12

ing 9.12 shows a complex object--
a building. A section has been
cut through the nearby wing
so that both interior and
exterior details are pictured.

REFERENCES

[1] Smith, Max J., Stephen L. Macdonald, C. Stephen Carr, Robert Wehrli, Edward F. Smith, Jay A. Schadel, David A. Luther, SPACE-FORM--Computer-aided Design for Architecture, Technical Report 1-2, University of Utah, Salt Lake City, Utah, September, 1968.

[2] Ezra Ehrankrantz has developed a systems approach for the design of schools and other buildings. This approach restructures the organization of prefabricators, contractors and buyers (school districts). The systems approach discussed herein is not antagonistic to this idea, but it does not require it. Its primary use is to organize the traditional design disciplines in building, along with their body of knowledge and theory. For a treatment of systems design and design methodology, see the following:

Roe, P.H., Soulis, G. N., Handa, V. K., The Discipline of Design, The University of Waterloo, October, 1965.

Wehrli, Robert, Open-Ended Problem Solving in Design, Department of Psychology, University of Utah, August, 1968.

[3] The integrated civil engineering system (ICES) as proposed by Daniel Roos and others at MIT is an integrated design system for

civil engineering. Certain of the system processes--COGO for geometric design and STRESS for structural design, for example--have been widely employed, but comprehensive computer systems combining these and other subsystems have not been realized. Except for the geometry in COGO, ICES relies upon the input of numbers and text rather than drawings.

Roos, Daniel, ICES System Design, The M.I.T. Press, Massachusetts Institute of Technology, Cambridge, Massachusetts, 1966.

[4] In his thesis project, David Luther developed a partial design system for electrical elements in buildings. Design was divided into three subsections; power supply, power distribution and lighting distribution. The designer could select a number of electrical parts, such as switches, convenience outlets, etc., and place these on a plan of the building displayed at the CRT. The electrical elements could be placed, re-placed and deleted at will. As they were placed, a record was maintained by the computer to permit an automatic print-out of all electrical elements and their attributes, such as wattage, manufacturer or installation height.

In this example, the architectural plan existed first in the project file of the ARCAID system and was transmitted to the electrical design system. This constituted a substantial test for transmitting data from one part of the comprehensive system to another.

Luther, David A., Electrical Design System, Master's Thesis, University of Utah, Salt Lake City, Utah, 1969.

[5] For a typical structural system, see STRESS, a part of the ICES proposal.

STRESS--A User's Manual, Department of Civil Engineering, Massachusetts Institute of Technology, M.I.T. Press, 1967, Fourth Printing.

STRUDL is an expanded and updated version of STRESS.

[6] Basic ideas for spaceforms as topologically described elements were originated by Ivan Sutherland and presented in SKETCHPAD. This development was extended to three-dimensional elements as noted in the following works:

Sutherland, Ivan E., SKETCHPAD: A Man-Machine Graphical Communication System. Technical Report No. 296, Lincoln Laboratories, Lexington, Massachusetts, January, 1963.

Johnson, Timothy. SKETCHPAD 3: Computer Program for Drawing in Three Dimensions," AFIPS Spring Joint Computer Conference, Detroit, 1963, 347-353.

[7] For a thorough treatment of perspectives in architectural use see the following:

Martin, C. Leslie, Architectural Graphics, The Macmillan Company, New York, 1952.

[8] Stephen Coons has developed systems for dealing with three-dimensional curved surfaces. He has also developed transformations by matrix algebra. This is a mathematical technique which permits rotation of objects as sequential transformations on lists of the points, lines, and planes of a graphic element.

Coons, Stephen, Surfaces for Computer-Aided Design of Spaceforms, Document #MAC-TR-41, Cambridge, Massachusetts: Massachusetts Institute of Technology, June 1967.

Coons, Stephen, "An Outline of the Requirements for a Computer-Aided Design System," Simulation, Vol. II. Issue 2, 1964, R-2.

[9] Rovner, Paul D. and Feldman, Jerome A. The Leap Language and Data Structure. Technical Report, Lincoln Laboratories, Lexington, Massachusetts, October, 1967.

[10] John Warnock and Gordon Romney have developed hidden line algorithms, and Stephen Carr is implementing these in ARCAID. It is anticipated that these algorithms can be operative in real time. This development will allow an object rotated constantly to appear with hidden lines removed.

Warnock, John E., A Hidden Line Algorithm for Half-tone Picture Representation, Technical Report 4-5, University of Utah, Salt Lake City, Utah, May, 1968.

Romney, Gordon, Computer Assisted Assembly and Rendering of Solids, University of Utah, Salt Lake City, Utah, 1969.

[11] Necker, Louis Albert, "On an Apparent Change of Position in a Drawing or Engraved Figure of a Crystal." In Dember, William N. (Ed.), Visual Perception: the Nineteenth Century. John Wiley & Sons, New York, 1964.

[12] In Machine Perception of Three-dimensional Solids, Larry Roberts explored the question of determining from a photograph the geometric nature and extent of physical objects.

Roberts, L. G., Machine Perception of Three-dimensional Solids, Technical Report No. 315, Lincoln Laboratories, Lexington, Massachusetts, May, 1963.

[13] Adams, Myron W. and Bradley, Prentice, A62 Guide for Modular Coordination, Modular Service Association, Publishers, 110 Arlington St., Boston, Massachusetts, 1946.

[14] The New Mathematics Dictionary and Handbook was used as a guide in arriving at some of the definitions. In nearly every case, however, the definitions have been adjusted as appropriate to this text.

Marks, Robert W., The New Mathematics Dictionary and Handbook, Bantam Books, New York, 1964.

APPENDIX A

Building Systems

A.1.1 Introduction

A building may be characterized as a set of systems. Typical building systems are site, acoustics, structures, enclosure, color, safety, mechanical, and appurtenances. The functional systems operate as a whole to carry out a more or less complete function in a building.

Functional systems are partitioned into object and attribute systems. Object systems are made up of objects which function systematically for a given purpose. Attribute systems are formed from the characteristics, or attributes of the objects.

In its completed form ARCAID will include a process for each of the object systems and many of the attribute systems in a building. A process consists of data and the procedures for managing or operating upon this data. This report uses the structural system to illustrate a typical building system. Input, analysis, and output are discussed.

A.1.2 Building Systems

A building is made up of many elements combined in a variety of ways. Enclosure elements define and enclose space; structural elements support the building against live and dead loads. Still other elements provide for the comfort of the building occupants with adequate lighting, heating and cooling, ventilation and acoustics. The function of an element determines the system or subsystem under which it is classified.

Some objects may perform several functions at the same time and be classified in more than one attribute system. For instance, a wall may provide visual and acoustical privacy as well as thermal insulation and protection from the weather.

A system is a group of elements having a common purpose or function, or a regular interaction. The interdependence of the elements and the function, use or output of the system are often more important than the elements taken by themselves. On the one hand, systems may be split into subsystems, assemblies, subassemblies, components and parts--systems being the largest element and parts of smallest. On the other hand, systems may be the components or subassemblies of larger entities called supersystems or hypersystems. The finished building is a supersystem because it is composed of several building systems.

The systems approach to building design allows the designer to design in a comprehensive manner. ARCAID is a proposed computer graphics system of hardware and software permitting an architect to design a building from briefing and schematic design through the construction document phase. ARCAID uses the SPACEFORM graphics language and a series of processes--one for each of the object and attribute systems. The SPACEFORM language gives the designer great spatial design freedom and provides the needed capabilities for picturing his schemes. The building systems processes provide him with (1) data, and (2) highly interactive simulation, synthesis, and analysis procedures for each of the building systems.

A procedure instructs the computer to perform a specific task or series of tasks. A subroutine is a procedure for a specific task,

and a routine is an integrated group of one or more subroutines. Each process combines a number of procedures--routines or subroutines--which permit the execution of a complex and extended task.

Object and attribute systems will now be described in greater detail.

A.1.3 Object Systems

Each of the building systems represents a convenient collection of subsystems that perform similar or related functions. The list of object systems--site, structures, enclosure/space-use, mechanical, electrical, communications, and appurtenances--is useful for classifying a large number of subsystems under eight headings which are established by tradition in the building industry. Associated with each of these object systems is a group of building trades and a number of contractors, material suppliers, component manufacturers, etc. Within each object system are subsystems, which are small enough and at a level that can be readily handled in design.

Each object system is conceived of as an entity that incorporates the physical objects in the system, the dynamics of the input and output of its media, and the responses of people who use or are influenced by the system. Take, for example, a forced-air heating system. The furnace, fan, plenum, ducts, and registers are the objects; the medium is the air which is input to the furnace, filtered, heated, output to the ducts, and ultimately to the living space; and the responses are those of building occupants whose comfort is affected by the heating system.

Each of the subsystems and its components: object, media, and human considerations, provide design parameters to be considered within that particular building system process.

A complete listing of all of the object systems may be made for any building. A typical listing is as follows:

OBJECT SYSTEM	SUBSYSTEM
Site	Earthwork
	Paving
	Landscaping
Structures	Footings
	Foundation
	Superstructure
Enclosure	Walls
	Floors
	Ceilings
	Roofs
Space-use	Rooms
	Suites
	Wings
	Storeys
	Complexes
Mechanical	Heating
	Air Conditioning
	Ventilating
	Plumbing
Electrical	Power distribution
	Lighting
Communications	Telephone
	Intercom
	Computer
Appurtenances	Furniture
	Fixtures
	Equipment

In ARCAID, the objects of the object subsystems may be represented geometrically by the SPACEFORM language. Media will be represented by simulation methods using symbolic representations for moving air, etc.

The human responses will take the form of design parameters, or constraints, on the system such as temperature levels, air change requirements, or noise level limits.

By definition, systems have a high degree of interaction among elements within the system. They may also have some interaction with other systems at the interface between the systems. If the effect of these interface interactions is to interrupt or interfere with the functioning of either system, a conflict exists. If the interaction is helpful in the functioning of the two systems, a blend is achieved. An example of a conflict is the intersection of a forced-air heating duct and a steel beam. If the path of the duct cannot be altered to eliminate the conflict, the functioning of one or both of the systems will be hampered. An example of a blend is the use of hollowed-out tubes for heating or electrical ducts in a concrete floor slab. The loss of concrete near the neutral axis of the slab causes an increase in structural efficiency. Since duct and floor structures are integrated, the chance of conflict is reduced. Major objectives of the building systems processes in ARCAID are to identify and constrain against conflicts, and to maximize blends.

A.1.4 Attribute Systems

The object systems of a building contain all of its objects. But each object in the building has a set of attributes that are of concern to the designer. These include size, weight, cost, color, texture, etc. An exhaustive listing is not possible even for relatively simple objects. The attributes may form systems each taking its "parts"

from the attributes of objects. Like object systems, the attribute systems also perform functions in the building. These functions, however, may be more subjective and abstract than those of the object systems--as for example, with safety or cost--which still influence building users directly or indirectly.

The attribute systems may be quite concrete ones such as texture or color, or they may be more abstract ones such as social aspects or human comfort. They may be readily quantifiable such as heat transmission, area, or volume; or they may be difficult to quantify such as aesthetics.

An exhaustive listing of attribute systems is not possible for any design project. They are selected from a great many which are potentially available. Those of concern to the designer will vary for each project he undertakes. A typical listing of attribute systems for a building is as follows:

ATTRIBUTE SYSTEM	SUBSYSTEM
Area	
Volume	
Cost	Initial cost Maintenance Building useful life
Materials	
Visual	Color Light (intensity, distribution) Reflectance Texture (visual)
Acoustics	Sound transmission Sound reinforcement Reverberation time
Thermal Characteristics	Heat transmission Expansion - Contraction

Safety	Fire resistance Fallout radiation protection
Tactile	Texture Rigidity Vibration
Weight	
Use Flexibility	
Aesthetics	
Ecology	
Social Aspects	

The function performed by an attribute system may be negative or undesired rather than positive and desired. Conflicts are likely to occur between attribute systems; but because they are less concrete and measurable, they may be less obvious than those that occur between object systems. An example of a conflict between attribute systems is a room with adequate area and seating capacity but which has poor sound reverberation time for its intended use. An example of a blend is a prefabricated wall panel that not only provides sound and thermal insulation but harmonious color, fire resistance, and low cost as well.

The simulation, synthesis, and analysis procedures for the attribute systems are similar to those for the object systems. The building systems processes assist in making the design decisions. The machine performs routine calculations and increases the designer's understanding of the building and its systems through graphic display of information, dynamic simulation, and a continual interaction and transfer of information between man and machine. The computer graphics

console provides access to the machine, SPACEFORM is the graphics language, and the building systems processes provide the design capabilities for each building system.

A.2 The Structures System Process

A.2.1 Introduction

Among the object system processes in ARCAID is one for structural design. It provides synthesis and analysis procedures to handle most, if not all, of the structural problems that may be encountered by the designer. It also provides structural data such as "tables" of rolled steel members. The structures process is developed by means of the SPACEFORM language.

The structures procedures will ultimately include analysis for tension, compression, torsion, shear, deformation, displacement, bending, buckling, stability, mobility, and collapse. Analysis of any one or of any combination of the above will be possible. The analysis procedures will handle individual structural elements such as beams, columns, struts, connectors, walls and footings; complex structures and structural assemblies including trusses, frames, shells, plates and tension structures; and, composite structure such as shells on columns, plates on frames and shear walls in frames. The most commonly used and available procedures are programmed first and others added as funding and time permit. The structural system may be independent of the enclosure system as for a steel frame, or it may be integral with the enclosure system as for a concrete or masonry bearing wall.

Structural design will be possible in any combination of the traditional structural materials of concrete, steel, and wood as well as newly developed materials such as plastics, reinforced resins, structural foams, or other non-ferrous metals. Again, expected demand will determine the priorities for programming.

The architect or structural designer uses the same graphics console with its peripheral devices as described in the body of this report. Using the SPACEFORM language, the designer can build up a scheme for the entire structural system, or for any subsystem or part. He is able to display the structural system independently or in conjunction with the enclosure or other building systems in order to detect blends or conflicts between systems.

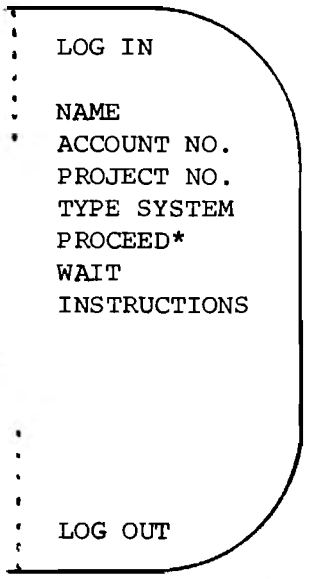
The designer might display only the structural frame on the CRT. He may then display the mechanical system overlaid upon the structural. A coding of linework permits him to distinguish between the two object systems. For example, the structural system might be shown with brighter lines than the mechanical system. Conflicts between ducts and beams are thus easy to identify.

The display is dynamic, allowing the designer all the housekeeping, shaping, assembling, viewing, orienting, and miscellaneous manipulations described previously. For example, the designer can point, place, or erase loads, structural members, joints and constraints. He can display stress distribution in individual members, joints or total structures as well as points of overstress. This might be shown with isostatic lines on each member of the structure or by some other appropriate means. Deflection, perhaps exaggerated for ease of inspection, or points of excessive deflection or displacement are shown.

Geometrical and topological information are transferred from the display to the analysis subroutine and back automatically, or as commanded by the designer. This interaction between designer, dynamic display, and sophisticated analysis subroutine provides great freedom and reduces the tedium of structural system design, redesign and optimization.

A.2.2 Input for Structures

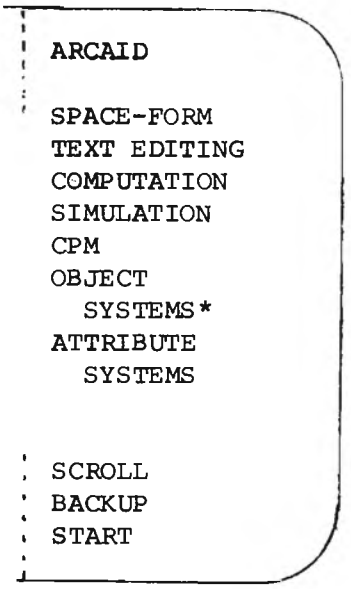
A typing keyboard is part of the designer's console upon which he may type appropriate commands. A more rapid means of interaction, however, is the display of commands in the key space of the CRT display, which the designer may select with the stylus. A tree structure is used to organize the structures process and its sub-elements. This permits moving from the general to the specific (or from the higher level data to the lower level) as well as moving along or between branches.



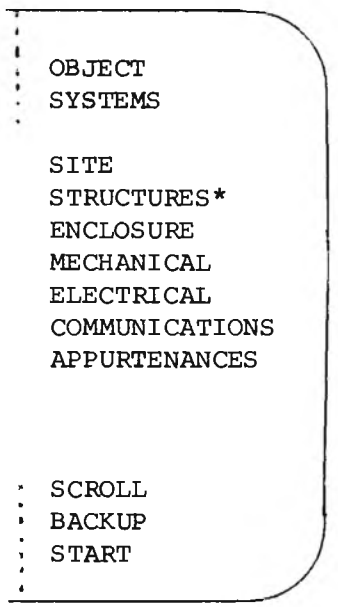
A.1

When the designer approaches the console, he logs in with his name and identification number (Fig. A.1). If he is unfamiliar with the operation of the system, he may engage in an interactive learning process with the machine by

asking for instructions. If he has used the system before and is familiar with its operation, he may select the broad and specific capabilities that he desires to be operational; he does this by selecting "Proceed" in the key. The operational processes are then displayed. He may then begin by choosing "Object Systems" from the key (Fig. A.2). The next display in the key will be a listing of the available Object Systems as shown in Fig. A.3. He may then select "Structures" and receive a display of the types of structure that the system is capable of handling (Fig. A.4). If he selects "Linear," he will receive a display indicating such considerations as loading, materials, input, output, and alternate paths he may follow (Fig. A.5). Each selection from the key brings up a new display of information, constraints, symbols, etc., whichever is applicable. The key also serves as a checklist for the designer indicating areas he has yet to consider and decisions he must make before proceeding. Typical command, information, and constraint displays are shown in Figure A.6.

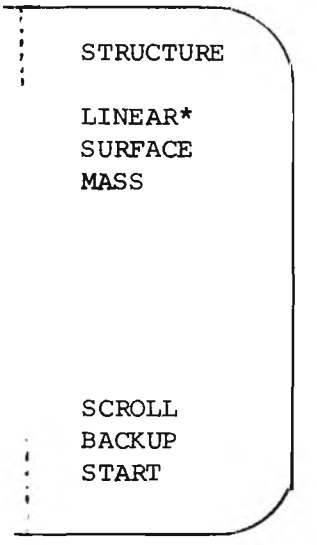


A.2



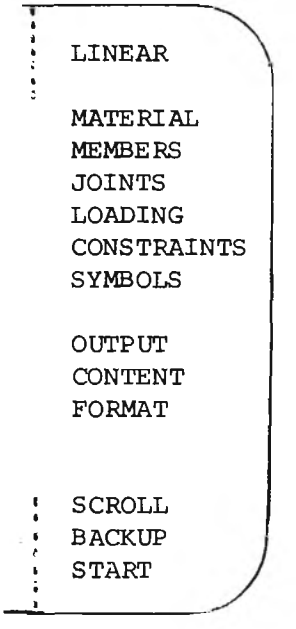
A.3

Near the bottom of the key in each display a series of commands appear. These allow the designer to move back and forth through the "tree" of data. Since any listing may contain more items than will fit into the key space, he may point the command "Scroll" and rotate the track-ball to scroll up or down the list. He may use the "Backup" command to retrace the path that he has followed or return to the point of beginning by pointing "Start." he may begin again or log out as he desires.



A.4

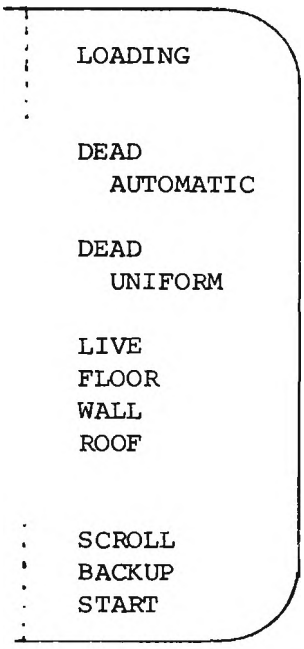
The designer may approach a structural design problem in several ways. Two such approaches will be described here. In the first method the designer begins by first building up the structural system, a bay frame building for example. He does so by assembling "frame" spaceforms, each representing a structural bay in his design. (Drawing A.7). When complete, he has a topological description of the structure displayed as a three-way grid.



A.5

In the second approach the designer begins by building up the enclosure elements

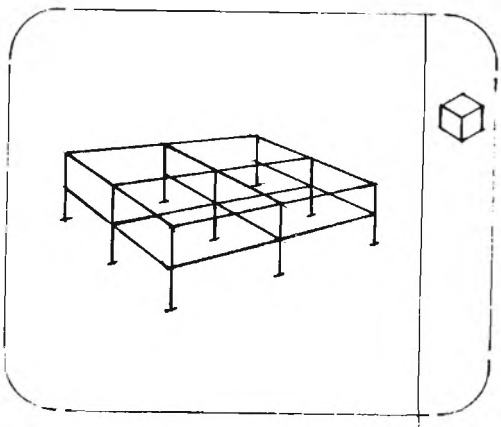
of his design using "box" or solid" spaceforms. (Drawing A.8). Enclosure system elements such as walls, floors, ceilings, windows, etc., may then be located and the spaces within the building thus defined as in Drawing A.9.



A.6

The designer most likely has a structural system in mind although he has yet to indicate its type, location, and material. He may, however, have left the edges of some enclosure elements visible for later use as a structural grid. In any event, he must now identify those enclosure elements, such as bearing walls, which contribute to the structure system and supply new structural members where needed.

Whether he begins with a consideration of the structural or enclosure systems, the designer must at some point, then, identify a complete system of structural support. He may then estimate the dead and live leads and position them throughout the structure for a preliminary analysis. If the design process involves objects from the enclosure system process, the weights of enclosure

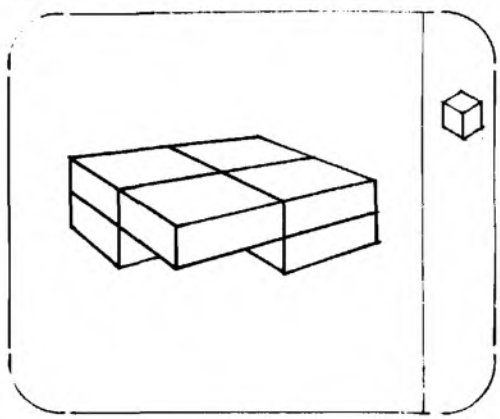


A.7

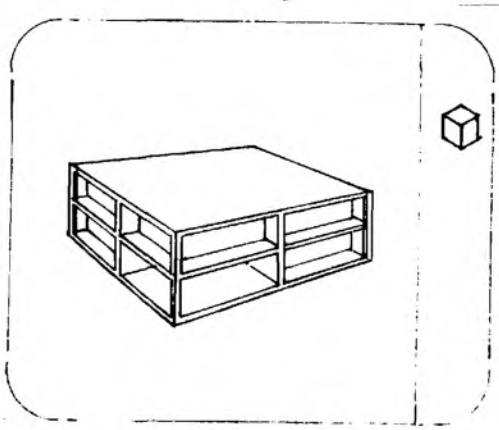
elements may be assigned automatically working through the attribute system processes.

The designer then continues to indicate joint characteristics, such as fixed or pinned, and any special features of the structural system.

The geometry of the structure (joint and member coordinates) as well as the topology (member connectivity) are input and become stored as part of the structural data. The designer must also designate the material of the structure (steel, concrete, etc.) and provide all additional constraints: for example, a restriction that only wide flange sections may be used or other limitations as to member sizes or deflections. He must also indicate the type and extent of any checking to be done--code checking, for example.



A.8



A.9

The designer may now invoke the analysis procedure that suits the level of complexity that he desires. A routine then performs a completeness check to ensure that sufficient information has been input for the wanted level of analysis. If the information is complete,

it transfers the geometrical and topological description and all parameters required to the analysis procedure for computation.

A.3 Analysis Procedures

The analysis procedures are software programs, specifically designed to perform rigorous analysis of particular structural systems or subsystems, designed with various structural materials. The designer must have a basic knowledge of structural behavior and a working knowledge of the capability of various structural systems and structural materials.

None of the presently available analysis procedures permit graphic input and output, which is the fundamental contribution of ARCAID and its SPACEFORM language. It allows the designer to interact with the machine in a real-time design and analysis process. The designer builds up his scheme at the display and the machine displays the results of its computations in graphic or alphanumeric form.

A.4 Output

The output in the structural process may take several forms. The designer may see directly the results of a particular analysis by a display of a member sizes and an indication of the stress and deflection in the individual members, perhaps as an exaggerated deflection. Any changes made in the structure, or its loadings, are reflected in a continually updated display.

The structural process also permits the plotting of hard copy in the form of conventional drawings such as framing plans, elevations,

isometrics and perspectives. This includes complete fabrication, assembly, and erection drawings at a level of detail now found in shop drawings.

APPENDIX B

Definitions

ABSTRACT MODE — Design with spaceforms which have extent and shape and perhaps a name, but no other attributes; contrasted with the concrete mode.

ANGLE — A primitive consisting of two intersecting lines. The angle degrees can be set by touching between the lines and typing the desired number of degrees. The lines assume the typed angle and retain it until a new number of degrees is set.

ARCAID — ARChitect's Computer Graphics AID. A computer graphics system of hardware and software for aiding the architectural designer in design. It is envisioned as a working environment which permits the complete description of a building without use of paper.

ASSEMBLE — To topologically combine primitive spaceforms or objects which need not then be proximate.

ASSEMBLING MANIPULATIONS — Those manipulations by which elements are topologically combined.

ATTRIBUTE — A specific quality or property which can be assigned to a scheme or parts of it.

ATTRIBUTE SYSTEM PROCESS — Those data and procedures applicable to one attribute system of a building.

AXIS — A line which has been designated as an axis by a name.

BASIC SPACEFORM — Any topologically described, simple, geometric solid such as a cube, cylinder, sphere, hexagonal rod, or triangular rod. Basic spaceforms have unit dimensions and occupy an area about one inch square in the key.

BOUNDARY CURVE — A curve on the face of a spaceform and used to define a curved surface.

BOUNDARY DEFINED SURFACE — A surface developed by defining boundary curves. The resulting surface has the shape of a perfectly elastic membrane stretched between the boundary curves.

BOX — A spaceform which is defined by opaque faces. A hole cut in a face of a box reveals the interior space.

BUILDING SYSTEM PROCESS — A Building System Process is specifically oriented toward architectural design. Building

System Processes are of two types: object system processes and attribute system processes. For any building, there is a small and exhaustive set of object systems. In ARCAID, the object system processes are as follows: site, structure, enclosure/space-use, mechanical, electrical, communications, and appurtenances. Since the attributes of objects are as broad as human experience itself, a great many attribute systems are ultimately possible. For the present, however, only the following attribute system processes are contemplated: area, volume, cost, material, visual, acoustics, thermal, and safety.

CENTROID — The center of visual mass of an element.

COMPONENT — An element made up of two or more parts.

CONCRETE MODE — Design with parts and components selected from object system processes; contrasted with the abstract mode.

CONSOLE — The computer graphics designer's work station.

CONSTRAIN — To limit the degrees of freedom available to a scheme. To constrain is to impose a decision which then affects later decisions.

CONSTRAINT — A specific storage representation of the relationship between elements which limit the freedom of the elements; i.e., reduces the number of degrees of freedom of the system.

CONTRACT-EXPAND — To contract, expand, or otherwise shape a graphic element, touch the corner or edge which is to be affected, and execute. Then move the corner or edge to the wanted location and execute. In some positions of an element such as a spaceform, corners, edges, and faces may be ambiguous; that is, an edge may be seen as a point, or a face may be seen as a line. In such cases, rotate the element prior to shaping to overcome the ambiguity.

CONVERGENCE — In perspective, the apparent angling of lines or edges as they recede from the eye.

COPY — Data stored in context with its process as, for example, data stored with the scheme in the project file; copies are contrasted with instances.

CORNER — A point which defines the intersection of two or more edges of a spaceform or object.

CURVE — A graphic element defined by its end points and equation; a curved line segment. The two types of curves are boundary and primitive curves.

CURVED SURFACE — An infinite aggregate of points constituting a space of two dimensions.

CUT — To cut an element (object) with a plane or curved surface. Cutting is done at right angles to the display plane. To cut, rotate the element into the desired position relative to the display surface and retrieve a plane or curve from the key and place it. If the purpose of the cut is shaping, touch the unwanted portion of the element (object), execute, and type the word DELETE. If the purpose of the cut is viewing, touch the portion of the element (object) to be removed, execute, and type the word ERASE. To restore an object cut for viewing to its original configuration, rotate the element to its position when cut, touch the curve or plane, execute, and type the word ERASE. This will erase the cutting curve or plane and restore the element to its original configuration.

CUTTING PLANE — Any plane or curved surface used to slice through any graphic or building element. Cutting may be for shaping or for viewing. For shaping, the unwanted portion is deleted. For viewing, the obscuring portion is temporarily removed from the picture, but remains in the scheme.

DELETE — To remove both from the picture and from the scheme. Deletions become garbage.

DESIGN — The man-machine process of producing a scheme.

DIMENSION — To assign an arrow and a value to the length of a line or edge. The arrow and value can be labeled.

DISPLAY PLANE — An imaginary plane surface taken to be approximately coplanar with the display surface.

DISPLAY SURFACE — The CRT surface available for displaying text, numbers, or graphics.

DRAW — To draw a curve. To draw, type the word DRAW and then, using the stylus, trace the desired curve, which appears on the working surface as it is drawn.

DRAWN CURVE — A curve which results from freehand drawing movements of the stylus; as contrasted with the primitive curve which is selected from a set of curves and then adjusted as needed.

DUPLICATE — To make a copy or an instance of an element.

EDGE — A line which defines the intersection of two faces of a spaceform or object, or the extent of a surface.

ELEMENT — Any textual, numeric, graphic or building part, component, or system.

ENLARGE — To increase the size of a displayed picture. To enlarge, press the "enlarge-reduce" button and move the footpedal with the toe. For enlarging, the station point and vanishing points may be thought of as part of the object such that an enlargment of the picture causes an equivalent enlargement of the imaginary "triangle" bounded by the station and vanishing points.

ERASE — To remove part or all of a picture from the working surface. To erase, touch the elements to be erased, execute, and type the word ERASE. If no elements are specifically touched, the entire picture will be erased from the working surface.

EXECUTE — To indicate to the computer that a decision has been reached and that the command should be carried out. To select or place, the execute command is accomplished by pressing down on the stylus to activate the switch in its tip. Execute commands for typed operators take effect when they are fully typed and the carriage return button is depressed.

EYEPOINT — The assumed position of the designer's eye relative to the display surface.

FACE — One of the surfaces which bounds a spaceform or object.

FALSE EDGE — A line, not an edge, which distinguishes a surface or other element from its ground. In most perspective views of a cylindrical spaceform, for example, one end appears as a distorted ellipse, and the opposite end as a segment of a distorted ellipse. These are joined by two straight lines which delimit the cylindrical surface from the ground. The straight lines are false edges. If the cylindrical spaceform is rotated on its axis of symmetry, the false edges retain their relationship of figure to ground, while constantly changing their relationship to the curved surface.

FIT — To apply a curve fitting routine to a drawn curve. To fit, touch the curve, execute, type FIT, and execute again.

FLOP — To duplicate an element which is reversed from left to right-hand from the original element.

FORESHORTENING — In perspective, the reduced apparent dimension of a line or surface as it is turned away from a frontal position.

FRAME — A spaceform which is defined by its edges only. Edges at the rear of the frame are visible unless concealed by forward edges.

GARBAGE — Computer data which is no longer wanted.

GENERAL PURPOSE PROCESSES — The general purpose processes are: text editing, computation, graphics, simulation, and diagnostics. These are always accessible to the computer graphics user and are not oriented toward any specific discipline such as architecture.

GRID — To superimpose a net on a surface.

GROUND — The background portion of the display which is not part of the object pictured.

HIDDEN LINE — An edge of a spaceform or object turned away from the designer or otherwise obscured by other spaceforms or objects such that it should not appear in the picture.

HOUSEKEEPING MANIPULATIONS — Manipulations general to ARCAID, or supportive of other manipulations.

INDEX — The first level in the tree structure of a given library; for example, the spaceform index contains representations of the basic spaceforms and primitives.

INSTANCE — Data not stored in context with its system; for example, data relative to a scheme and accessed by a pointer rather than stored with the scheme in the project file. Instances are contrasted with copies.

KEY — Elements displayed on the key surface. Typical keys are lists of elements or commands available for designing schemes on the working surface.

KEY SURFACE — The righthand portion of the display surface used for displaying keys.

LABEL — To display names and attributes. To label, touch the elements to be labeled, execute, and type the word LABEL.

LIBRARY — Data stored and available for general problem solving or for designing a building scheme. In ARCAID, the library is partitioned according to processes and then further organized in an heirarchical (tree) structure.

LIBRARY, ARCAID — The store of those elements in ARCAID which are not allocable to one or another of the processes.

LINE — A graphic element defined by its end points; a straight line segment.

LOCATION, 2-D — A point on the image of a graphics element or on the display surface or both.

LOCATION, 3-D — A point on a graphic element.

MAKE COLINEAR — To make two lines colinear, first touch one line with the stylus, execute, type COLINEAR, touch the second line, and execute. The first line will move to a position colinear with the second. The lines may be edges of spaceforms or building parts.

MAKE COORDINATE

To make two points coordinate, first touch one with the stylus, execute, type COORDINATE, touch the second point and execute. The first point will move to a position coordinate with the second. The points may be the corners of spaceforms or objects.

MAKE COPLANAR

To make two surfaces coplanar, first touch one surface with the stylus, execute, type COPLANAR, touch the second surface, and execute. The first surface will move to a position coplanar with the second. The surfaces may be faces of spaceforms or building parts.

MANIPULATION

Changes in pictures affected by the hand or foot movements of the designer.

MOVEMENT

Hand or foot movements of the zoom pedal, track ball, stylus, light pen, mouse or push buttons as well as typing.

MOUSE

A device for manipulating a tracking cross on the display surface.

NAME

To assign text to a graphic element. The text can be a name, or one or more attributes. To name,

touch the element, execute, type name or attributes, and execute again. The text will appear as a label at the point of stylus contact until the next command is given; at which time, the label will disappear.

NET — Two families of parallel lines perpendicular to each other on a surface.

NET NODE — An intersection of lines of a surface net. By touching nodes with the stylus and pulling up or down, the surface can be re-shaped.

NUMERIC CONTROL — A computer process for driving machine tools or other machinery for making real world objects.

OBJECT — Anything topologically described in ARCAID. Objects are displayed on the key and working surfaces. Spaceforms are abstract, having only shape and size. Objects are more concrete and have additional attributes. At the display, we see pictures of objects, but name them and otherwise deal with them as if they were the objects themselves.

OBJECT SYSTEM PROCESS — Those data and procedures applicable to one object system of a building and available to the designer in ARCAID.

ORIENTING MANIPULATIONS — Those manipulations needed to visualize one element with its parts, with other elements, with a site, or with coordinate planes.

OUTLINE — The sum of edges and false edges, or segments of them which delineate a graphic element from its ground.

PACK — To place spaceforms or building parts in components without voids which are "unassigned;" for example, to pack rooms of a building. Touch and execute for each element to be packed, then type PACK. When the packing involves certain constraints such as minimal dimensions for certain elements, these constraints are input prior to packing.

PAN — To represent the viewing of an object achieved by rotating the viewer's head. To pan, press the "pan" button located near the track ball, and rotate the ball. For panning, the relative positions of the station and vanishing points remain constant, but they move in a circular arc with the station point as the axis of revolution. Objects at the same distance from the viewer retain the same relative size.

PART — The smallest element of a system which is of concern to the designer.

PERSPECTIVE — The "camera" view of an object involving convergence and foreshortening.

PICTURE — A representation of points, lines, and surfaces on the display surface. Pictures may include text, numbers, and graphics.

PLACE — To place an element topologically.

PLANE — A graphic element that wholly contains every straight line joining any two points that lie within it.

PLOT — To plot a picture of an element.

PLOTTER — A machine for making two-dimensional hard copies.

POINT — A graphic element having only a location in space.

POSITION — The position of a graphic element or object in three-dimensional space.

PRIMITIVE — The point, line, curve, net, and surface. Primitives are available in the SPACEFORM index.

PRIMITIVE CURVE — The circle, ellipse, parabola, hyperbola, and catenary. Primitive curves are available in the graphics index.

PROCESS — Any associated procedures and data which aid the designer in accomplishing a given task. General purpose processes permit the use of words, numbers, and graphics in a broad range of problems or activities. Object and attribute processes aid in more specific design tasks with particular relevance to architectural design.

PROJECT — A building to be designed or being designed.

PROJECT FILE — A large storage block which permits storage of a scheme in its "present" state. The scheme-as-stored does not affect the data structure of ARCAID; neither is the scheme-as-stored affected by those transformations of its data which are continuously needed for viewing manipulations and the like. A large number of schemes can be designed concurrently on the on-line consoles of a single, large computer. A "blank" storage

block on which the description for a scheme is made.

RECOVER — To delete the last command given. To recover, type RECOVER and execute. Recover will delete a command in process if given before the final "execute" of such command.

REDUCE — To decrease the size of a displayed picture. To reduce, press the "enlarge-reduce" button and move the footpedal with the heel. For reducing, the station point and vanishing points may be thought of as part of the object such that a reduction of the picture causes an equivalent reduction of the imaginary triangle bounded by the station and vanishing points.

RETRIEVE A SURFACE — To retrieve a curved surface once all boundary curves have been drawn or fitted to a given spaceform. To retrieve, touch the surface area, type RETRIEVE SURFACE, and execute.

REVOLVE — To develop a space form of revolution. To revolve, retrieve a line from the key and label it as an axis by typing the word AXIS. All revolution is done in the surface plane or in a plane parallel to it. Therefore, rotate the axis until it appears

as a point. Retrieve the wanted element for revolution and place it relative to axis, execute, and type the word REVOLVE.

ROTATE

To rotate a part or all of an element (object), press the "rotate" button located near the track ball and rotate the ball in the desired direction. The element will rotate about a point near its centroid. If the designer wishes to rotate about any other point, he can select a point from the key, name it CENTROID, place it, execute, press the "rotate" button, and then rotate the track ball. To rotate a part of an object, touch the stylus on the wanted part or parts, execute, and rotate.

SCAN

To represent moving past an element (object). To scan, press the "scan" button near the track ball and move the track ball in the direction of simulated movement. For scanning, vanishing points retain a fixed relative position with respect to one another and the station point and vanishing point "move with the viewer." Objects retain the same apparent size during scanning, and the translation of the object is opposite to the simulated movement of the viewer.

SCHEME — A description for some project being designed; the description of a project at a given moment in time; also, an alternative scheme. The scheme is stored in the project file.

SCROLL — To bring portions of a picture, not previously visible, into view on the display, as in the case of text which can be scrolled into view continuously. This is accomplished by pressing the "scroll" button located at the track ball and then moving the track ball. Only orthographic projections, text, and numbers can be scrolled. Only up-down and side-to-side scrolling are possible.

SECTIONAL ZOOM — To zoom through the faces of an element so as to view its interior.

SHAPE A SURFACE — To reshape the interior of a curved surface by manipulating nodes of a surface net. To shape, touch each node at the periphery of the area which is to be reshaped and type the word FIT. Then touch in turn each node which is to be changed, execute, place, and execute again.

SHAPING MANIPULATIONS — Those manipulations which modify the form of an element.

SHAPED SPACEFORM — A basic speacform which has been modified.

SIZE — To assign an extent to a line or edge.

SOLID — A spaceform of solid matter. A hole cut in a solid reveals a pattern of points and lines to represent solid matter.

SPACEFORM — A 3-D topologically described graphic element made up of points, lines, curves, and surfaces. Spaceforms have only area and/or volume. When additional attributes are assigned, spaceforms become parts, components, or systems of buildings and assume the names of such parts, components, or systems.

SPACEFORM LANGUAGE — A graphic language for manipulating graphic elements at the CRT.

SPACEFORM OF REVOLUTION — A shaped spaceform which has been created by revolving a graphic element about an axis.

SPOT — One of the bright dots on the display used to indicate that the computer has recognized which element the user has touched. It should not be confused with a point. A spot is placed by means of the tracking cross.

STATION POINT — In perspective, the assumed position of the eye relative to the object and picture plane; not to be confused with the eyepoint, which is the position of the designer's eye relative to the display surface.

STORE — To remove an element from the working surface and place it in storage. Storage may be in a general purpose, object or attribute process, in the ARCAID library, or in the project file. To store in the project file, first touch the element, execute, label, execute, and type STORE. To store other than in the project file, it is necessary to type the storage process prior to typing the word STORE.

SURFACE — A plane or a curved surface.

SURFACE NET — A 2-D grid of rectangles which may be applied to any surface. A surface net permits re-shaping of the surface by moving net nodes.

SYSTEM — Any group of elements Which has or can be conceived of as having a high density of internal relationships. In general, the internal relationships are more dense than the external relationships between the system and adjacent systems.

TRACK BALL — A device for translation, rotation, and perspective manipulations.

TOPOLOGY — Connectiveness--the means by which elements are related in the data structure; for example--points are connected with lines to form more complex figures.

TOUCH — To identify an element.

TRACKING CROSS — A small cross on the display surface indicating the active location of the light pen, stylus, or mouse.

TRANSLATE — To move an element in the display plane or in a plane parallel to the display plane. To translate, touch the element with the stylus, execute, move the stylus to a new location, and execute again. The station and vanishing points of the translated element are those of the untranslated portions of the object, which remain constant. Therefore, the convergence and foreshortening of the translated object constantly change with respect to the fixed points during translation.

VANISHING POINT	In perspective, an imaginary point to which parallel lines or edges of an object appear to converge. Vanishing points are not used in the perspective algorithms, but rather as a means of discussing convergence, foreshortening, and related concepts in perspective.
VIEWING MANIPULATIONS	Those manipulations which permit visualization of an element but do not modify it.
VISIBILITY	The availability of procedures for review by the designer.
WORKING SURFACE	The main portion of the display surface used for designing a scheme.
ZOOM	To represent moving toward or away from an element pictured.
ZOOM PEDAL	A device for zooming toward or away from a pictured spaceform or object.