DOCUMENT RESUME

ED 113 833 EA 007 637

AUTHOR	Eastman, Charles; And Others
TITLE	An Outline of the Building Description System. Research Report No. 50.
INSTITUTION	Carnegie-Mellon Univ., Pittsburgh, Pa. Inst. of Physical Planning.
REPORT NO	RR-50
PUB DATE	Sep 74
NOTE	23p.
EDRS PRICE	MF-$0.76 HC-$1.58 Plus Postage
DESCRIPTORS	*Architectural Drafting; *Architectural Programing; *Building Plans; *Computer Graphics; *Computer Programs; Facility Planning; Facility Requirements; Spatial Relationship; Systems Approach

ABSTRACT

Many of the costs of design, construction, and building operation derive from the reliance on drawings as the description of record of the building. As a replacement, this paper outlines the design of a computer system useful for storing and manipulating design information at a detail allowing design, construction, and operational analysis. A building is considered as the spatial composition of a set of parts. The system, called Building Description System (BDS) has the following associated with it: (1) a means for easy graphic entering of arbitrarily complex element shapes; (2) an interactive graphic language for editing and composing element arrangements; (3) hardcopy graphic capabilities that can produce perspective or orthographic drawings of high quality; and (4) a sort and format capability allowing sorting of the data base by attributes, for example, material type, supplier, or composing a data set for analysis. (Author)

An Outline of the Building Description System

by

Charles Eastman, David Fisher,
Gilles Lafue, Joseph Lividini,
Douglas Stoker, Christos Yessios

Institute of Physical Planning

Research Report No. 50

September, 1974

Abstract

Many of the costs of design, construction, and building operation derive from the reliance on drawings as the description of record of the building. This paper outlines as a replacement the design of a computer system useful for storing and manipulating design information at a detail allowing design, construction, and operational analysis. A building is considered as the spatial composition of a set of parts. The system, called Building Description System (BDS) has associated with it: (a) a means for easy graphic entering of arbitrarily complex element shapes; (b) an interactive graphic language for editing and composing element arrangements; (c) hardcopy graphic capabilities that can produce perspective or orthographic drawings of high quality; (d) a sort and format capability allowing sorting of the database by attributes, e.g. material type, supplier, or composing a dataset for analysis. The system runs on a Digital Equipment PDP-11/20, with extended disc memory and graphics.

This report is a progress report, outlining the goals and current status of work on BDS.

I. The Problem

To date, the only practical medium for communicating the information needed for construction of buildings is drawings. Drawings are also the principal medium for communicating with all other parties involved in a building project e.g. client, building inspectors, and financial institutions. The necessity for drawings results from the spatial information that must be conveyed. Of course, drawings are augmented by notes and written specifications.

Architectural drawings have many inherent weaknesses. They are highly redundant, describing the same part of a building at several different scales. Since drawings are two-dimensional and a building three-, at least two drawings are required to characterize any part of the building arrangement, and on these, at least one dimension must be depicted twice. Thus information on drawings is inherently redundant and a design change leads to changes in a whole set of drawings. A large amount of effort is directed at keeping the various drawings consistent.

A large effort is also directed at keeping current the information in the set of drawings for a building project. But even with this effort, at any moment, at least some of the information depicted by a drawing is not current or not consistent. Thus decisionmaking by one group of designers may often be based on obsolete information, further complicating their task.

While drawings are currently uniquely useful for dealing with the spatial arrangement of buildings, numerically represented information is required for most analyses. Currently, this information must be manually taken off construction drawings. This initial step of data preparation, at present, is the major cost in any building analysis.

After construction, changes to a building are always depicted on separate sets of drawings (to distinguish current work from past). Thus drawings accumulate and after the initial construction, it is the rare building which exists for which a single integrated set of drawings exist describing its current state. Thus a building ages, the recorded information about its physical arrangement deteriorates.

II. Conceptual Design of a General Building Description System

BDS was initiated to show that a computer-based description of a building could replicate or improve on all current strengths of drawings as a medium for building design, construction and operation as well as eliminate most of their current weaknesses. Our premise was that a computer database could be developed that would allow the geometric, spatial, and property description of a very large number of physical elements, arranged in space and "connected" as in an actual building. Conceptually, the model would be similar to a balsa wood model, but with far greater detail. In addition, spaces as well as solids would be explicitly depicted. The database would provide a single description of each building element or space, relative to others, and thus would allow any change to be described only once rather than copied onto a large number of drawings. The elemental parts of a building would be drawn in by the user or stored in one or more libraries of components. Thus there would be no restriction as to the range of designs possible. On the other hand, this one database could easily handle all industrial or prefabricated building systems as well as buildings composed of custom or on-site fabricated components.

An important feature of the BDS model is its capability for generating drawings. From this single database, the designer could ask for any plan or section, perspective or exploded view and receive construction detail documents of high quality in a short period of time and at low cost. All drawings produced from the same database would be automatically consistant.

In a similar vein, because the building description is now in a machine readable form, any type of quantitative analysis could be directly coupled to the system. All data preparation for such analyses would be automatic, greatly reducing their cost.

With such a database, qualitative analyses would be similarly facilitated. Perspective drawings of any view of the exterior or interior of the building would be available on both drawings or on a cathode ray tube (crt) display. Both line drawings and half-tone displays could be available. Visual inspection should be greatly enhanced, due to the infinite range of views available.

In addition, building code checks on this database have the potential of being automated and violations could be checked for during design regularly. During construction, programs for producing various shop drawings could be utilized. Quantity take-offs and parts lists of mechanical and other fabricated parts could be done automatically. Later, the computer database, on magnetic tape, would be useful for evaluating building operations, such as mechanical equipment cycles. With appropriate flagging with dates, this database would also be useful for later remodeling and renovation work throughout the building's life.

Preliminary studies have indicated that a broad variety of design philosophies could be easily accommodated by such a system. A designer could start a project with structure, spaces to be enclosed, or geometrical system with equal ease. Currently, no operations or data base are agreed upon by design or building professionals, primarily because there has been little need for such consensus forming. The availability of BDS gives reason for developing a consensus. We anticipate BDS to incorporate low level operations which will be both necessary and sufficient for most operations required within prevailing design philosophy and that the operations will be structured into a convenient language for both direct use and composition into more complex operations.

Many of the above proposed uses of a building description database have large implications for practices in the design and construction fields.

It is not the purpose here to argue for each of these changes, but to point out the broad range implications of a BDS. Given the drafting and analysis efficiencies during design alone, it seems reasonable to expect a BDS to reduce the direct costs of design, as now carried out, by more than fifty percent.

In order to depict a building at construction detail as an arrangement of parts, we must have some estimate of the number of parts that must be described. Great variation will exist in practice, but for purposes of estimation we assume that a primitive part to be described in such a database is an element that comes to the building site as a separate unit, using on-site construction. Thus we believe screws and clips might be eliminated from consideration, though not plumbing valves or pipe hangers.

In Table One, we show the assumptions used to derive an estimate for the total number of parts for a ten storey building of about 120 thousand square feet of floor area. It suggests that a building of this size is likely to have about 150,000 separate parts. Design variations lead to at least an order of magnitude variation in this number for a building of the same size.

A reasonable target figure then, for a production oriented general BDS system, is for it to hold 200-400 thousand elements. Given that the appropriate technology today for mass storage is a disc-pack (typically sized at twenty-million words), this amounts to 80 to 160 words of memory per element, when program space is subtracted. This number provides a target figure for the database design, given the state of computer hardware technology available to architects and engineers.

III. Realization of a General Building Description System

Realization of a BDS involves solving at least the following technical problems. Some have been resolved earlier by others; other problems have already been solved by us. Others remain to be resolved. The problems whose resolution are not obvious include:

A. identification of a hardware configuration that is economical and available to design professionals.

B. design of a data structure capable of allowing "full" descriptions of any three dimensional solid objects or spaces. This description must allow complex shapes, surface description, tests of inside-outside, and a wide range of attributes and relational data.

C. incorporation of the data structure into a generalized database allowing on the order of magnitude of two hundred thousand elements to be individually described and manipulated, within the hardware configuration proposed.

Table One

Prototypical Building (Office Type)

	parts:
size:	
100' x 100' by 10 storeys = 100,000 sq. ft.	
plus two basements = 20,000 sq. ft.	
exterior walls:	
4 ft panels, 8 parts/panel	
400 linear feet of wall/floor	
1000 panels x 8	8,000
basement walls:	
one part/panel	200
interior walls:	
5 linear feet/100 sq. ft. of floor area	18,000
3 parts/linear ft. $\frac{120,000}{20} \times 3$	
floor and ceilings:	
1 part/sq. ft.	120,000
general equipment:	
10,000 misc.	10,000
total parts	156,200

or 1.25 parts/sq. ft.

D. design a specialized executive program which is fully compatible with and knowledgeable about the database of BDS. The executive must provide the interface bewteen BDS, its host hardware and the user.

E. given such a large spatially oriented database, means must be developed to quickly sort elements of interest from the total set.

F. entering of so many elements is also an issue. One facility needed is for easily entering complex three-dimensional shapes.

G. an equally important facility is required to efficiently arrange large numbers of (potentially similar) physical elements.

H. needed also is an easy means for editing an arrangement, including both the shape and location of an element, or sets of similar elements.

I. a set of general manipulation routines are required, particularly for the computation of shapes and for interrogating the database.

J. a facility for generating high quality displays of subsets of the database, for inspection or editing.

K. a similar but extended facility for producing high quality architectural drawings of different parts of the modeled building.

L. a report generating facility, for quantity surveys and parts schedules, as well as for preparing databases for analytic programs.

M. incorporation of the above operations into a formally organized and easily understood man-machine language.

Most of these technical issues listed above have been addressed and resolved already. Those remaining are viewed as tractable. In the following section, our treatment of each of the above technical issues are outlined.

IV A. Hardware

Two candidate hardware configurations for BDS are office resident minicomputers and time-sharing access to a large central computer. The BDS basically consists of a very large database and routines to manipulate it. This database must reside in close proximity to the cpu which operates on it; any other arrangement would result in inordinate communication costs and time. Other desirable features include real time generation of graphical displays of the database and easy switching from one database, i.e. building project, to another. These features, plus the speed and long term cost advantages of minicomputers, has encouraged us to follow the mini-computer line of development.

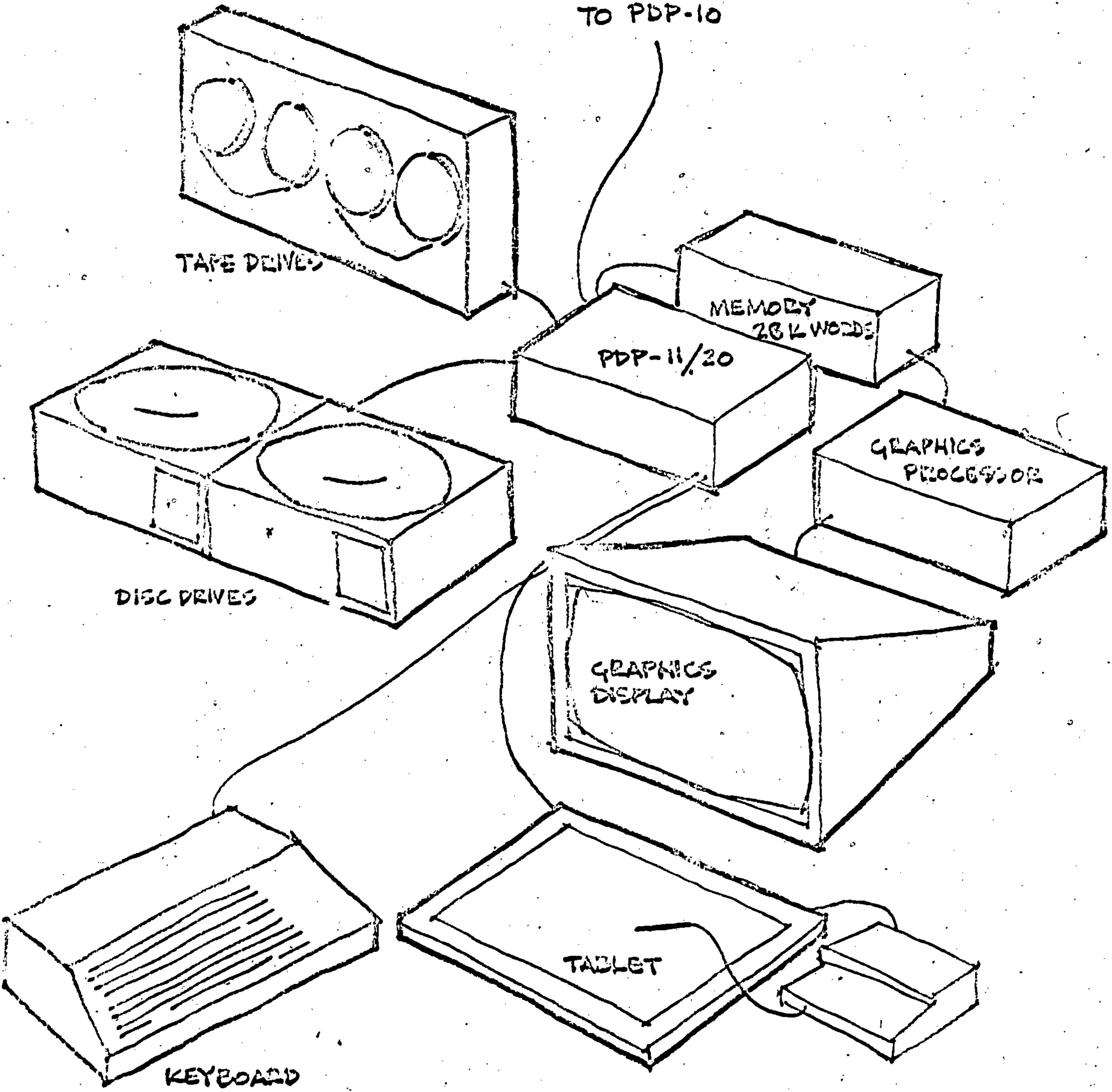

TO PDP-10
TAPE DRIVES
MEMORY
28K WORDS
PDP-11/20
DISC DRIVES
GRAPHICS
PROCESSOR
GRAPHICS
DISPLAY
KEYBOARD
TABLET

FIGURE A.1.

Sixteen bit mini-computers are becoming increasing common which operate at micro-second speeds. These machines commonly have addressing facilities for at least 32 thousand words, with fast communication with disc memory extending this to about 60 million words. At current prices, a hardware system comprised of a PDP-11 (40-45) with 28K words, one twenty million word disc, and a refresh (or more cheaply, a storage tube type) graphic display can be purchased for about fifty thousand dollars. This amounts to the equivalent payroll of one draftsman distributed over a life of three to five years. As computer prices continue to drop, this relation will become even more attractive. To us, this cost could be easily justified today by an appropriate design tool. It is on these assumptions regarding computer hardware that we are proceeding.

Our hardware configuration is shown in Figure A.1. The graphics processor is a custom build unit of high capability described in a paper by Rosen.* The only difference between our prototype configuration and that proposed is a smaller amount of disc storage, currently 2.5 million words.

IV B. Data Structure for Shapes

The BDS data structure is organized similarly to the topology of any geometric solid. That is, a body is made up of vertices, edges and faces and the relations among these parts are as defined in the classical works of geometry of Klein and Mobius.** The structure of the database is shown in Figure B1. Our inspiration for the database draws significantly from the work of Baumgart.***

This database specifically provides the geometric capabilities for depicting all examples of closed solid polyhedra, with planar faces. By closed, we mean that the edges of all faces are adjacent to other faces. By solid, we mean that only one side of a face may be considered the exterior of the body it is part of. It should be noted that certain geometric solids do not satisfy these conditions; Mobius bands and Klein bottles are not solid in this sense and thus cannot be represented in our data structure (but these are not meaningful architectural shapes anyway). In general, all figures must satisfy Euler's equation relating the number of faces, edges, and vertices of a polyhedron, "vertices minus edges plus faces equals two times the number of bodies". For our work, faces have been limited to plane surfaces. Each shape can be dimensioned to seven places decimal accuracy (25 bit mantissa, 8 bit exponent). The properties easily accessed for each shape by simple operators include its overlap condition with any other shape, the relation of any point to a face (inside, or, outside) and the relation of a point to a shape (inside, surface, outside).

---

* Brian Rosen, "The Architecture of a High-Perofmrance Graphic Display Terminal" International SID Symposium Digest, May 1973, New York, New York.

** F. Klein Geometry: Mathematics From an Advanced Standpoint Vol. II Dover Publication, 1939.

*** B. Baumgart, "Winged Edge Polyhedron Representation" Stanford A.I. Report No. AIM-179, Computer Science Dept. October, 1972.

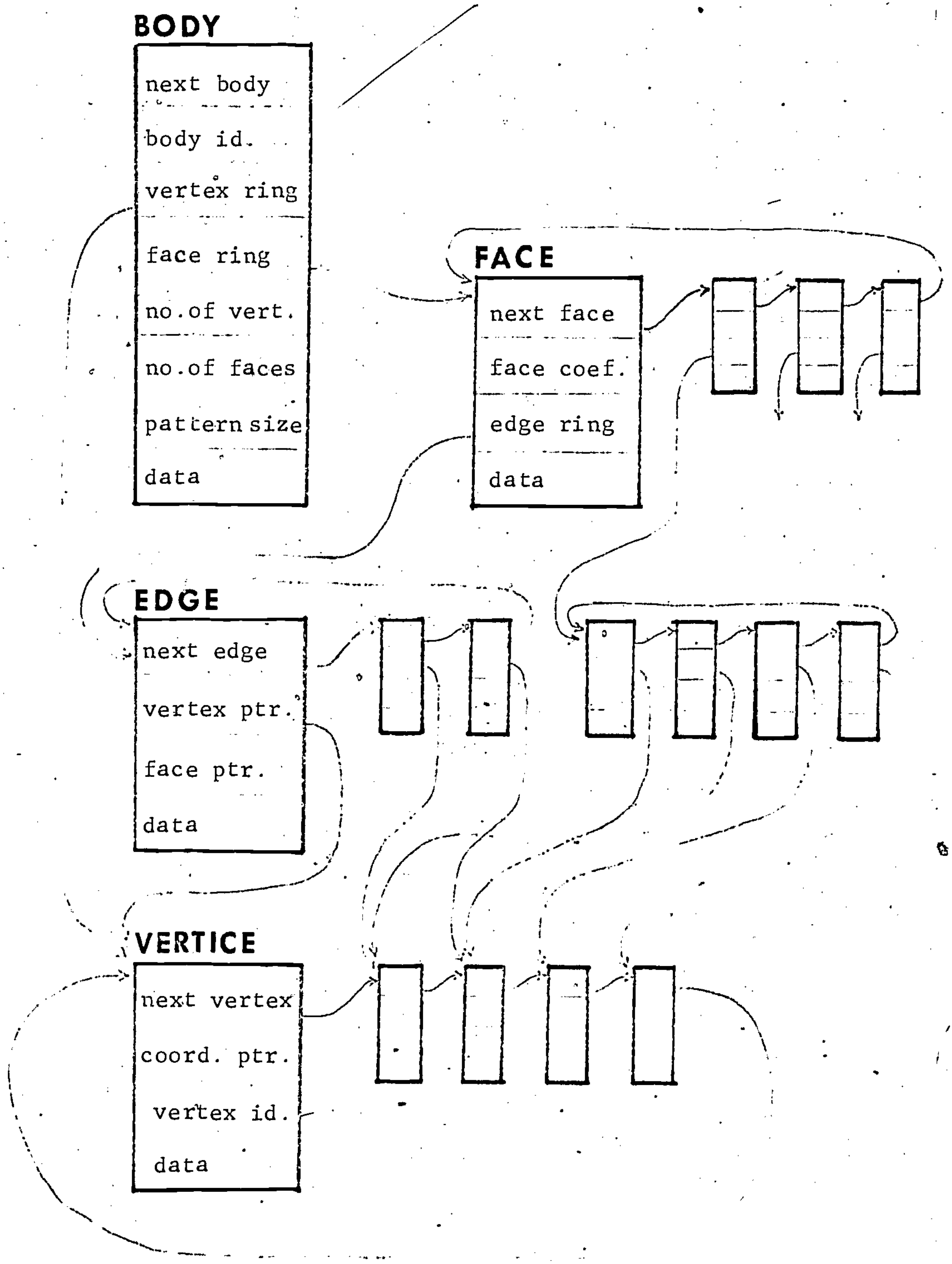


FIGURE B1
Data structure for depicting the topology of an element shape

An important feature of the data structure is that it is based on a set of efficient operators for building shape descriptions. A series of operations are available, corresponding to the set of equivalencies in the Euler equation. The result of each operator is an extension or deletion to the data structure which is logically consistant with the Euler equation. Thus the Euler equations provide a discipline very useful in defining three-dimensional solids.

The data structure for shapes has been implemented and tested. The details of the shape data structure are presented elsewhere.*

IV C. The Data Base for Storing and Accessing Large Numbers of Elements

It would be inefficient to store the shape of every element in a building if it is repeated in different locations. Moreover, many shapes are highly repetitive if the dimensions of the shape are not considered. The properties of a shape remaining after dimensions are ignored are its topology and that is what is incorporated in the shape description outlined in the previous section. In this sense, all wide flange beams have the same shape, all pipe and tubing have the same shape, etc. Moreover, with elements of the same shape but different dimensions, the potential location of vertices vary in systematic ways. All windows from a company's line may be of the same shape and vary only in size, as defined in two dimensions. Correspondingly, the detailed geometric description of this window requires only these two dimensions, if the window was initially described as a topology combined with a set of expressions for relating each coordinate location to the two given dimensions. The same is true for wide flange structural steel (defined by five parameters), tubing and pipe (two parameters).

The above recognitions suggest that there are four levels of data in a shape description of a building element. These are graphically depicted in Figure C1. The four levels are called the pattern, expression, template, and instance levels, from top to bottom. The pattern level stores a shape topology as described in the previous section; the expression level stores the relation among all shape dimensions in parametric form; the template level stores the parameter values giving specific dimensions to a shape, and the instance level incorporates a location description in the building space, called a spatial transform. All but the bottom level in the hierarchy involves a one-many relation. Through this hierarchy, one pattern and set of expressions are capable of depicting all tubing and pipe, or all structural steel "H" and "W" shapes. The expressions give different dimensional proportions to a pattern. Thus one set of expressions may define all rectangles and another set of expressions for the same pattern may define all trapazoids, as in Figure C1. Proper combination of pattern, expression, and values results in the complete spatial definition of an element, but without a location (we say it is located at the origin). Another example of the use of the hierarchy is given in Figure C2.

---

* D. Stoker and C. Eastman "The Machine Description of Complex e-Dimensional Bodies" Institute of Physical Planning Report, Carnegie-Mellon University (in preparation).

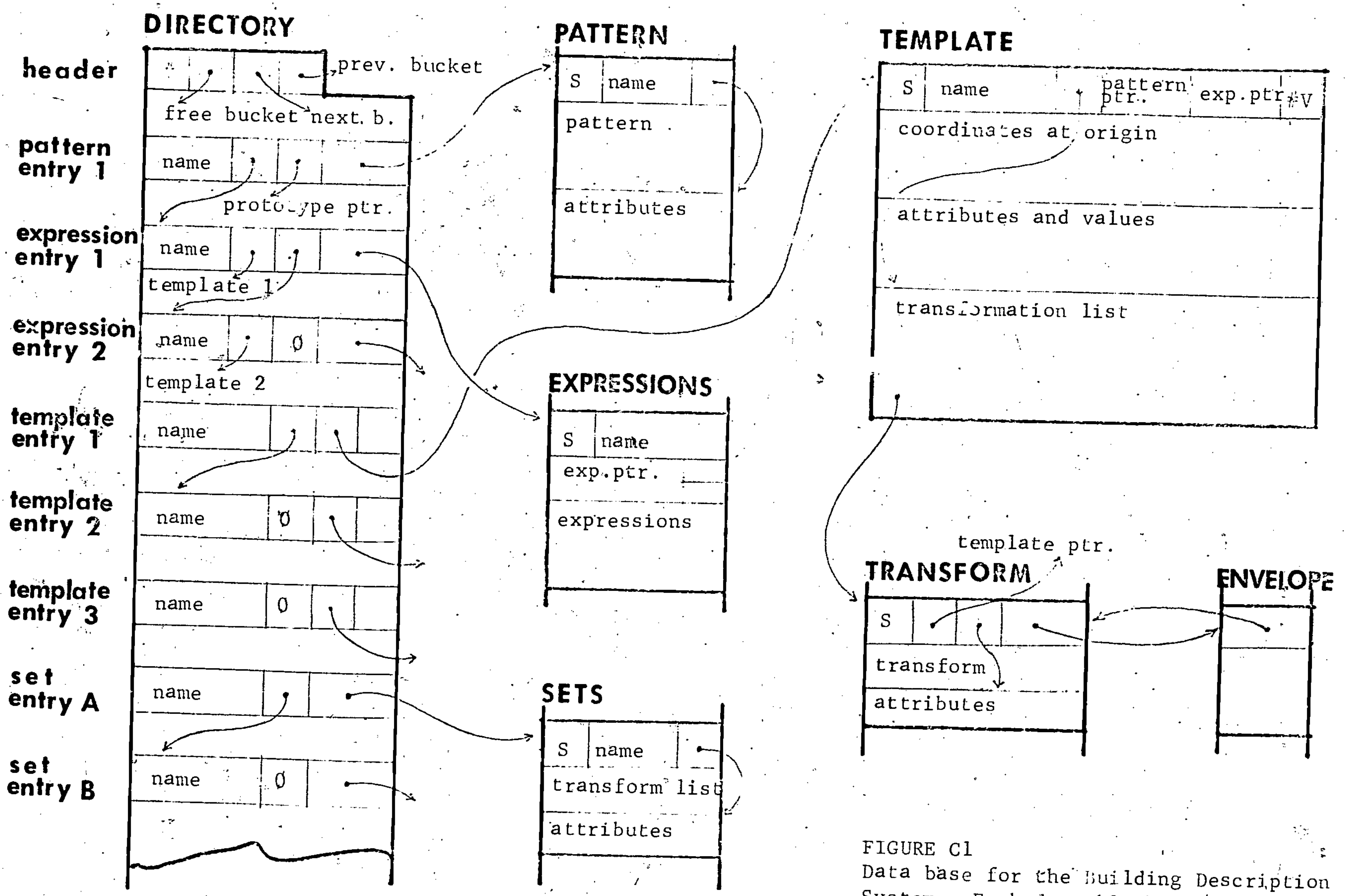


FIGURE C1
Data base for the Building Description System. Each data block is addressible disc and in core.

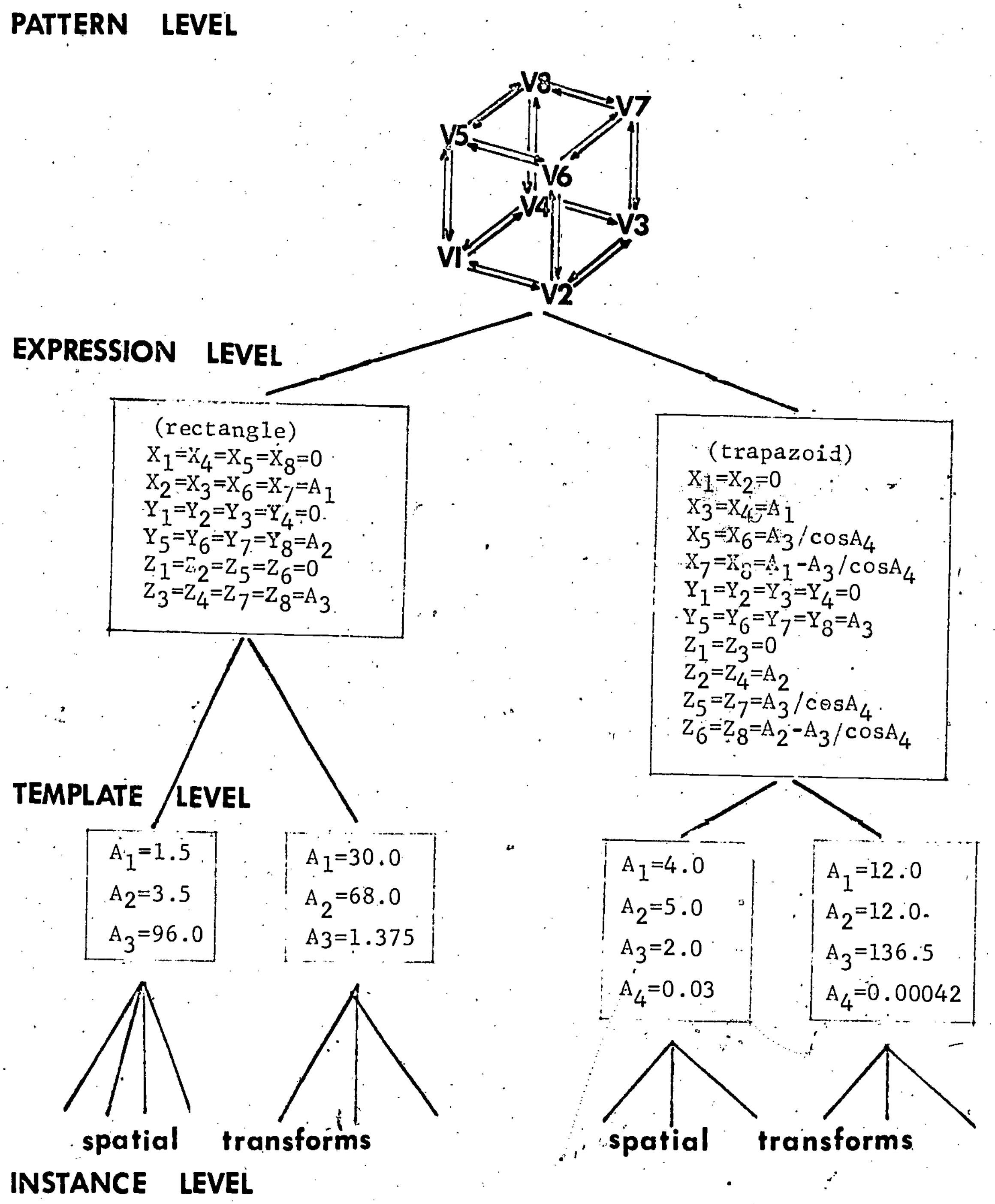
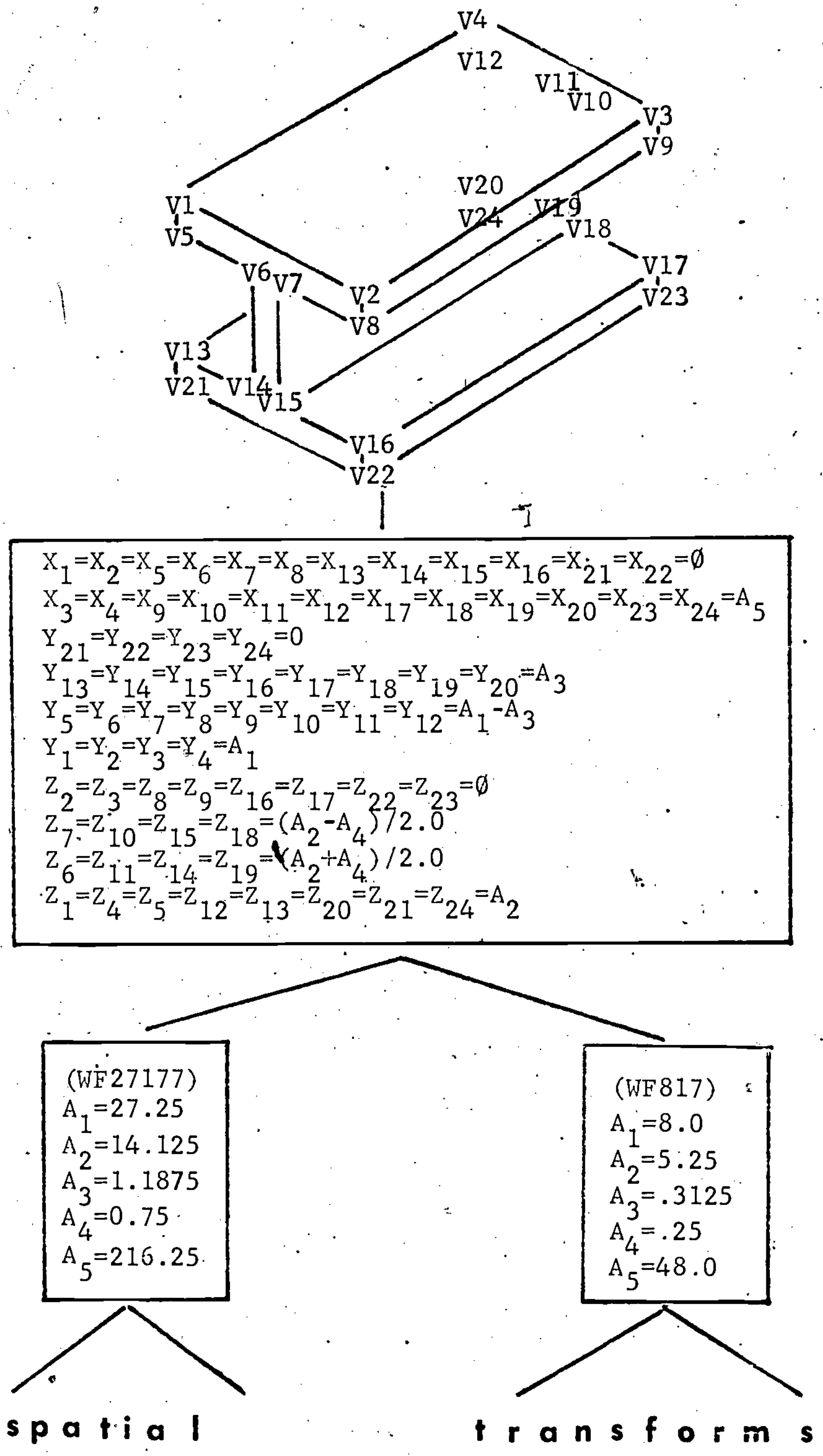


FIGURE C2
The four hierarchical levels within the BDS database. The shapes depicted are an eight foot 2x4, a door, a rubber motor mount, and a concrete column respectively.



$X_1=X_2=X_5=X_6=X_7=X_8=X_{13}=X_{14}=X_{15}=X_{16}=X_{21}=X_{22}=0$
$X_3=X_4=X_9=X_{10}=X_{11}=X_{12}=X_{17}=X_{18}=X_{19}=X_{20}=X_{23}=X_{24}=A_5$
$Y_{21}=Y_{22}=Y_{23}=Y_{24}=0$
$Y_{13}=Y_{14}=Y_{15}=Y_{16}=Y_{17}=Y_{18}=Y_{19}=Y_{20}=A_3$
$Y_5=Y_6=Y_7=Y_8=Y_9=Y_{10}=Y_{11}=Y_{12}=A_1-A_3$
$Y_1=Y_2=Y_3=Y_4=A_1$
$Z_2=Z_3=Z_8=Z_9=Z_{16}=Z_{17}=Z_{22}=Z_{23}=0$
$Z_7=Z_{10}=Z_{15}=Z_{18}=(A_2-A_4)/2.0$
$Z_6=Z_{11}=Z_{14}=Z_{19}=(A_2+A_4)/2.0$
$Z_1=Z_4=Z_5=Z_{12}=Z_{13}=Z_{20}=Z_{21}=Z_{24}=A_2$

(WF27177)
$A_1=27.25$
$A_2=14.125$
$A_3=1.1875$
$A_4=0.75$
$A_5=216.25$

(WF817)
$A_1=8.0$
$A_2=5.25$
$A_3=.3125$
$A_4=.25$
$A_5=48.0$

FIGURE C3

A single pattern and set of expressions are sufficient to define all wide flange beams. For each beam size and length a separate set of values are entered where A_1 is depth, A_2 is width, A_3 is flange thickness A_4 is web thickness and A_5 is length.

The data for describing a physical element according to this hierarchy will reside on disc. Fast access of this data during computation of an element is most important for quick response of the system.

The product of expressions and values is the coordinates of objects at the origin. These will only be modified occasionally and instead of computing the coordinates at the origin each time, we can store them at the template level. To compute an element in world coordinates what we call instantiate it, then requires three pieces of data, the pattern, the coordinates stored at the template level and the transform stored at the instance level. (Each of these may require its own disc access. This implies that the rate at which elements are "brought up" for use on our hardware will be about ten per second.)

Figure C3 depicts the database and its block structure on disc. In the current structure, once an object's expressions and values are defined, its coordinates at the origin are stored. Each level is stored as a separate data element.

A user must be able to conveniently enter a new pattern, new expressions for an existing pattern, or new values for an existing expression-pattern combination. Moreover, consideration must be given to editing and reviewing of existing patterns, expressions, and values already stored. Also in certain elements, there will be no need to define vertice coordinates through expressions and values because all instances will have the same shape and dimensions. In this case coordinates should be entered directly, without intervening expressions. These we call simple templates. All the features described are provided by the database, as shown in Figure C3.

Within the database, accesses to the different data elements are through a common directory. Each pattern, expression, and template have a unique name and entry. Within the directory, all expressions based on a common pattern are chain linked and all templates based on common expressions are linked. Also, those templates without expressions that are directly defined are linked to the pattern that they are associated with. These linkages duplicate the relations shown in Figure C1 and allow operations on those sets of elements related by the hierarchy.

In addition to the hierarchical generation of elements, we have provided for the naming and definition of sets of elements, allowing arbitrary forms of aggregation at the instance level of definition. Sets are envisaged for grouping classes of elements, e.g. structural, mechanical, or those provided by a particular supplier. Elements may belong to multiple sets and sets of sets are possible.

Associated with the pattern, template, and instance levels of the hierarchy are variable length attribute fields. These will be accessed and linked through an attribute directory, the details of which remain to be defined.

While the four-level hierarchy provides great flexibility in defining classes of elements, it could also be awkward to use in simple applications. Through the use of simple templates and careful design of the man-machine language, we are confident that the definition of elements will be both efficient and convenient.

The details of the database are presented in a separate report.* The database has been implemented and is now receiving preliminary testing

IV D. Spatial Search

Effort was made at the outset of the Building Description System project to find fast ways to access the database according to the spatial organization of elements. Particularly needed is the capability to access all elements overlapping a spatial area of interest. Some significant advances were made on this problem resulting in core oriented spatial searches being reduced by 50% or more and disc oriented searches requiring accesses to only those sectors holding elements of interest. Searching also allows accessing of elements to be based on the size of object, which is extremely useful for the generation of displays or drawings. The algorithms for core oriented searches have been tested and both classes of algorithms are presented elsewhere.**

IV E. Entering of Element Shapes

It is expected that descriptions of standard elements used in construction will be stored for BDS, most likely on magnetic tape Widespread use of BDS could result in companies providing information in the correct format as a service. In this case, the element need only be identified from a catalog and accessed.

If current architectural practices continue, it is imperative that a designer be able to enter elements of unique shape. One can anticipate a continuing need for defining such elements made of concrete, plastics, ductwork, and other locally fabricated materials. To date, there are only extremely tedious methods available for entering such shapes into a computer database.

---

* C. Eastman, J. Lividini, D. Stoker "A Database for Very Large Physical Systems", Institute of Physical Planning Report, Carnegie-Mellon University (in preparation).

** C. Eastman and J. Lividini "Spatial Search" Institute of Physical Planning Technical Report, Carnegie-Mellon University, September 1974.

Work is proceeding on the interpretation of elements of arbitrary shape, when traced on a tablet in two or three orthographic views. These views are entered sequentially, in a "natural" way. Given two views of most objects, there exists an inevitable ambiguity regarding the match of vertices. These ambiguities can be minimized, by: (1) skewing the orientation of the element; (2) decomposing the element into simpler parts, then gluing the parts into a whole, or (3) entering a third view. The system being developed by Gilles Lafue allows the user to utilize any or all of these "tricks" to intelligently describe an element shape to the BDS system.

When graphically entering an element shape, the system stores the corresponding pattern and its simple template. The user may add expressions and values if he desires.

After an element is entered, its simple template can be interactively adjusted so that the origin of the coordinates which describe it is in a convenient location for later mapping into the building space.

IV F. Placing Elements

Given one or a set of elements at the origin, a way is needed to locate a large number of instances throughout a building. This involves two kinds of issues: the composition of assemblages of elements at the origin so that a set can be relationally located in one step, and development of a set of iteration operations allowing repeated translations of the set mixed with rotations and placements.

The composition of assemblages is facilitated by the set definition of elements at the instance level A group of elements can be appropriately placed relative to one another with this facility. A general location operation, e.g. execution of a spatial transformation, can incorporate a multiply operation with the stored one to place the composition in the building space. Multiple instances of the sets are accommodated by the database. To facilitate bookkeeping, an element located as part of a set will not be able to be individually moved without an explicit "Breaking" of the set.

In a previous programming effort at the Institute of Physical Planning, a simple language of mathematical symmetry was developed for manipulating graphic databases. Called SYMPL (<u>Sym</u>metry <u>P</u>erspective <u>L</u>anguage), this program allowed definition of iterated calls of symmetry operators for application to sets of graphically described elements. It was found to be both powerful and convenient and will form the basis for the work on placing elements.

IV G. Arrangement Editing

Editing may involve changing the pattern of an element, its expressions, values, or transpose or a combination of these. Any change at some level in the hierarchy of generation must delete the instantiation information below it. Some thought was given in the organization of the database to facilitate editing; the expressions are stored in character string, algebraic form; values are stored, though used only once.

The specific operations to be used in editing have not yet been defined. This section of the system will probably be addressed in the Spring of 1975.

IV H. General Manipulation Operations

Preliminary analysis has shown that the following operations would be extremely useful in the interactive design of buildings:

1. the union, intersection, and differences of spatial domains;
2. sectioning of a form, as cut by some plane
3. deriving the space(s) enclosed by a set of solid elements
4. a pointing operation for identifying particular elements through interactive graphics;
5. a dilation operator which generates an element whose size is related to other elements

Some preliminary work on #1 and #2 have been undertaken by Douglas Stoker and Chris Yessios, with several different algorithms being identified. No choice has yet been made on the one to develop. Operations #4 and #5 are directly implementable, using available algorithms. No satisfactory algorithm has yet been proposed for #3. The power fo these operations were explored in an early paper.*

IV I. Graphic Displays

High quality interactive graphics will be a major form of man-machine interaction within the Building Description System. The initial work on displays has been undertaken by David Fisher, using a set of "drawing" routines implemented by Joe Lividini. Orthographic and architectural drawings are initially being treated as a special form of perspective. Orthographic views result when the viewpoint is a very large distance from the viewing plane.

---

*C. Eastman "An Interrogation Language for Building Description Systems" Institute of Physical Planning Technical Report, Carnegie-Mellon University, August, 1973.

We are relying on the standard method of perspective transformation, in which all elements are moved to the origin for their "picture" with a viewpoint transform.* Given that the templates of all elements are stored about the origin and a location is computed via a spatial transform, any view of an element can be derived by multiplying the viewpoint transform with the complement of the spatial transform. Any element is transformed only once. A temporary set of perspective vertices are then created and read through the pattern so as to result in a perspective image, which is sent to the display processor. Both format-free display routines and composeable ones are being implemented. These will eventually include hidden line removal, though they do not currently. Work on producing half-tone displays has been initiated by Angelos Boyiadgis.

IV J. Architectural Drawings

Later, we expect to focus on automatic composition of high quality architectural drawings. Among the issues we plan to investigate are automatic methods of dimensioning and automatic determination of line width and type. It is anticipated that drawings will be generated not from a perspective view with the viewpoint set at infinity but from the cartesian coordinates of sections of the elements of interest.

Work on this topic is expected to begin early in 1975.

IV K. Report Generation

This capability will be used to generate reports and prepare data for analyses. It will provide a general interface allowing already written programs to interface with BDS. No work has yet been initiated on this task.

IV L. A Building Description Language

A system for man-machine problemsolving is ultimately successful to the degree that a problemsolver will use the system to solve real tasks. A powerful facility for aiding him is necessary, but not sufficient. Also required is a convenient system that generates net benefits when the full costs of changing traditional modes of problemsolving are included.

We have been encouraged in the work thus far regarding two aspects of man-machine communication. First, it seems quite possible to evolve the manipulation and display operations on this database into a formal

---

* W. Newman and R. Sproul, Principles of Interactive Computer Graphics McGraw-Hill, New York 1973, Chapter 12.

interpretive language, with APL as a rough prototype. Similar to most languages, the expected BDS language will consist of distinct manipulation, and input-output operations. In our case, though, an integration of these operations into a common formalism is the challenge, due to the much greater emphasis in BDS of feedback and input-output.

Second, aspects of this language must explicitly deal with the graphic nature and size of the database. These are greatly facilitated by graphic operations. The important graphic operations we have identified thus far for inclusion into BDS are graphical identification of operands and interactive parameterization of some operations, e.g. the symmetry or perspective display operations. The implementation of these operations can, of course, vary according to the hardware available. Thus graphics operations seem to be easily included as syntactic units in the language.

A very important aspect of man-machine interaction is feedback while data elements are being defined or manipulated. In algebraic languages, interim feedback is facilitated by free format output. We have provided the equivalent, format-free display of building elements, parts of elements, or sets of them, during most of the operations within BDS.

Good computer programming naturally results in rationalization of the programming effort, including modularization and generalization of routines wherever possible. The logical extension of such thinking is the development of a set of operation primitives and data types that can be combined to create all the high level functions of the system. Such modularization greatly facilitates both programming and transfer to other hardware. Thus far, BDS has accepted a high degree of rationalization. Of course, only the full development of the system will determine the extent of rationalization possible. The combination of user primitives and machine primitives give us high hope for definition of a formal language of BDS operations.

By careful composition of these facilities, we expect BDS to be convenient to a knowledgeable user. We are not aware of any unambiguous intuitive or generally known language of design. BDS will, by default, incorporate an initial effort in this direction.

IV M. Executive Program.

The operating system and executive monitor for BDS had to respond to several conflicting factors. The graphics hardware included a monitor that provided good graphics handling and communication, but only rudimentary in-core space allocation. The graphics hardware used addressing schemes that precluded the use of any of the Digital Equipment operating systems. Most operating systems include features which are not necessary for our use, e.g. loaders, linkers, editers, as we may use these facilities in the host PDP-10.

The design considerations and features of the monitor implemented by us include minimal size, knowledge of the database, optimal execution of disk accessing, and dynamic core allocation with a primitive form of virtual memory. It incorporates many of the features of the graphics executive written by Don Bihary.*

V. Summary

The goal is to develop a computer database capable of describing buildings at construction detail and to develop a powerful set of operations for that database. Of course, the system outlined here could be equally used for the preliminary stages of design. It would also be useful for the design of many artifacts besides buildings.

Our orientation is not to provide specialized analyses or operation packages that make commitments to any particular philosophy of design. Rather, we are attempting to develop a very redundant system allowing many ways to define the same design, but each with different but meaningful side effects. With this general tool as a beginning, we expect others to develop more personalized extensions for private use.

The Building Description System is being implemented in the BLISS system building language developed at C-MU and now supported by DEC. A common version of BLISS compiles and executes on PDP-10s and all design and trial implementations are in this mode. There also exists a BLISS cross-compiler on the PDP-10 that generates MACRO-11 assembly code for the PDP-11. Thus debugged code can be directly transferred.

This work is supported by the National Science Foundation and the Advanced Research Projects Agency of the Secretary of Defense.

---

*D. Bihary, GLSTER SYSTEM, Department of Computer Science, Carnegie-Mellon University, Pittsburgh, PA, 12/1/73.