COMPUTER GRAPHICS AND IMAGE PROCESSING (1974) 3, (91–124)

Computer Graphics for Drafting

ROBERT SEDGEWICK*
Western Electric Engineering Research Center, Princeton, New Jersey

Communicated by A. van Dam

Received December 1, 1973

Despite the fact that one of the most obvious applications of computer graphics is as a drawing board for the creation of precision drafted output, graphics systems being used solely for this purpose are rare. The conceptually simple capability of drafting for the sake of drafting seems to have been lost (or absorbed) in the wake of much more exotic applications.

This paper describes a system which applies interactive computer graphics to the drafting of highly complex telephone office engineering drawings. The characteristics of the problem and its solution will be outlined, and some features of the design and implementation of the system will be discussed in detail.

1. INTRODUCTION

The use of interactive computer graphics as a "drawing board" has been with us from the beginning: the ability to enter pictures into the computer is one of the most fundamental concepts of computer graphics. However, one rarely encounters a system which is being used solely for the creation of drawings. In typical interactive graphics applications, either the interaction involves much more than drawing (e. g., computer aided design), or the information entered is used for much more than the production of a drawing (e. g., automated machining). The final words of the Sketchpad report by I. E. Sutherland (the first real treatment of the classical drawing problem) appear to have been particularly prophetic: "It is worthwhile to make drawings on the computer only if you get something more out of the drawing than just a drawing. . . . We are as yet a long way from being able to produce routine drawings with the computer" [74].

This paper will describe a system which was designed, developed, and implemented solely for the purpose of producing "routine" drawings. The drawings being produced are telephone office drawings, which are extremely complex, highly diverse, and are currently interesting only as drawings. The number of such drawings to be produced is virtually unbounded, and there is strong economic motivation for reducing the drafting time required for their preparation. The interactive graphics system was developed to be used as a tool by the draftsman (rather than to replace him),

* Current address: Department of Computer Science, Stanford University, Stanford, California.

allowing him to produce exactly the drawings he produces by conventional methods, but in one-fifth to one-tenth the time.

The following section will place this system in perspective with respect to past uses of computers for producing drawings. Section 3 will outline some of the specific attributes of the problem of drafting for telephone office engineering. Section 4 will describe the use of interactive graphics in the solution of this problem. In Section 5 some aspects of the design of the system will be outlined and Section 6 will offer some concluding remarks.

2. PERSPECTIVE

The classical attitude that drawing on the computer can be useful only if extra information is attached to the drawing, coupled with the fact that this can lead to great savings in many application areas, has meant that virtually all interactive computer graphics applications systems to date have been design systems rather than drafting systems. Large systems have been developed using interactive graphics for design in the automotive [33,34,69,78] and aerospace [10,30,39,54,58,59] industries: the classical examples of this are the General Motors DAC systems for automobile design and the Lockheed Georgia aircraft simulation and design systems [10,58,59]. More recently, there has been an emphasis on integrated circuit mask design [5,26,27,31,35,44,50,60,62,71,84], printed circuit board layout [3,38], and electronic circuit analysis systems [9,15,16,17,20,38,47,70]. Other examples of application areas where drawing or picture editing systems have been applied to aid in the design process are architecture [2,23,51], logic schematic drawings [49], and piping layout design [73].

It must be emphasized that these are design systems, and the effort in the research cited above has been to make the power of the computer available to the designer, with computer graphics used as a means of communication. Successful as these systems are for design, they are not applicable to the drafting problem. It has been generally conceded that the cost of doing pure drafting (if it has to be done at all) has to be absorbed in the design savings. The reason for this is that drafting is simply a different problem from design, and it must be recognized that some of the classical interactive graphics techniques (which are really based on the design problem) may not be oriented towards the drafting problem.

Although many of these design systems do produce drawings, they are in general byproducts of some other process. The problem of doing real drafting by computer is unimportant in these systems because a precisely dimensioned drawing of an object is not necessary if the precise computer description of the object can be made available to the next step in the design (or production) process. It is important to note that this may not always be the case; and the ability to do drafting alone by computer can sometimes be necessary. The application to be discussed in this paper is an example, and it represents a whole class of applications where precisely dimensioned "blueprints" are a necessary finished product. Another example is found in documentation systems, where it is often necessary to make large numbers of drawings which simply describe objects and processes, and relate not at all to their design.

At the other end of the spectrum is the approach to drawing by computer which has emphasized output. A large variety of devices exist for making very high quality drawings by computer control. However, they are generally addressed to the kinds of drawings which can be somehow generated by the computer. There was originally great optimism that a graphics "language" could be developed in which any drawing could be defined and then output on a precision plotter or microfilm device [42,43]. (There is continuing research into computer languages for describing pictures, and much of the work is at an abstract level beyond the scope of this paper. However, some examples of practical noninteractive description languages can be found in [7,11,26,62]; and [8,52,55] allow picture definition on an interactive basis.) This approach does have its applications ([30] is the most well-known; other examples are [6,32,48]), but for most drafting applications it becomes more difficult to prepare a drawing for computer generation than it is to simply draw it. The output devices are very useful for preparing high quality drawings but they are in general necessary and not sufficient for drafting by computer graphics.

In short, there appears to be no "total" system (either in the literature or in industry) providing interactive assistance to the draftsman in producing routine drawings. The systems are rather oriented towards either programmers or designers. Of course, there is a large body of work relevant to the design of a drafting system: the input devices and techniques of the classical drawing packages now used in computer aided design systems: and the sophisticated computer controlled drawing devices now used for precision output of computer generated drawings. This paper will give a case study of a system which combines these technologies for the solution of a specific drafting problem. The output technology is almost directly applicable, and it will be treated only lightly; the paper will be mainly concerned with the design and development of the drawing package and the man–machine interface where it was found that the specific requirements of drafting necessitated a reformulation of some of the traditional ideas.

3. CHARACTERISTICS OF THE PROBLEM

Almost all of the engineering activity for building new telephone offices and updating old ones revolves around the "floor plan" and related drawings, which precisely define every aspect of every component of the telephone office. These drawings are extremely complex, and the drafting effort required to produce them is massive; their preparation can take on the order of hundreds of hours by conventional methods. Not only are large numbers of such drawings produced, but also they represent the only reliable descriptions of the telephone offices and, as such, are the basic means of communication between a wide variety of individuals and organizations. Consequently, there exists a complex system for controlling the production and use of the drawings. The problem under consideration here is the devel-

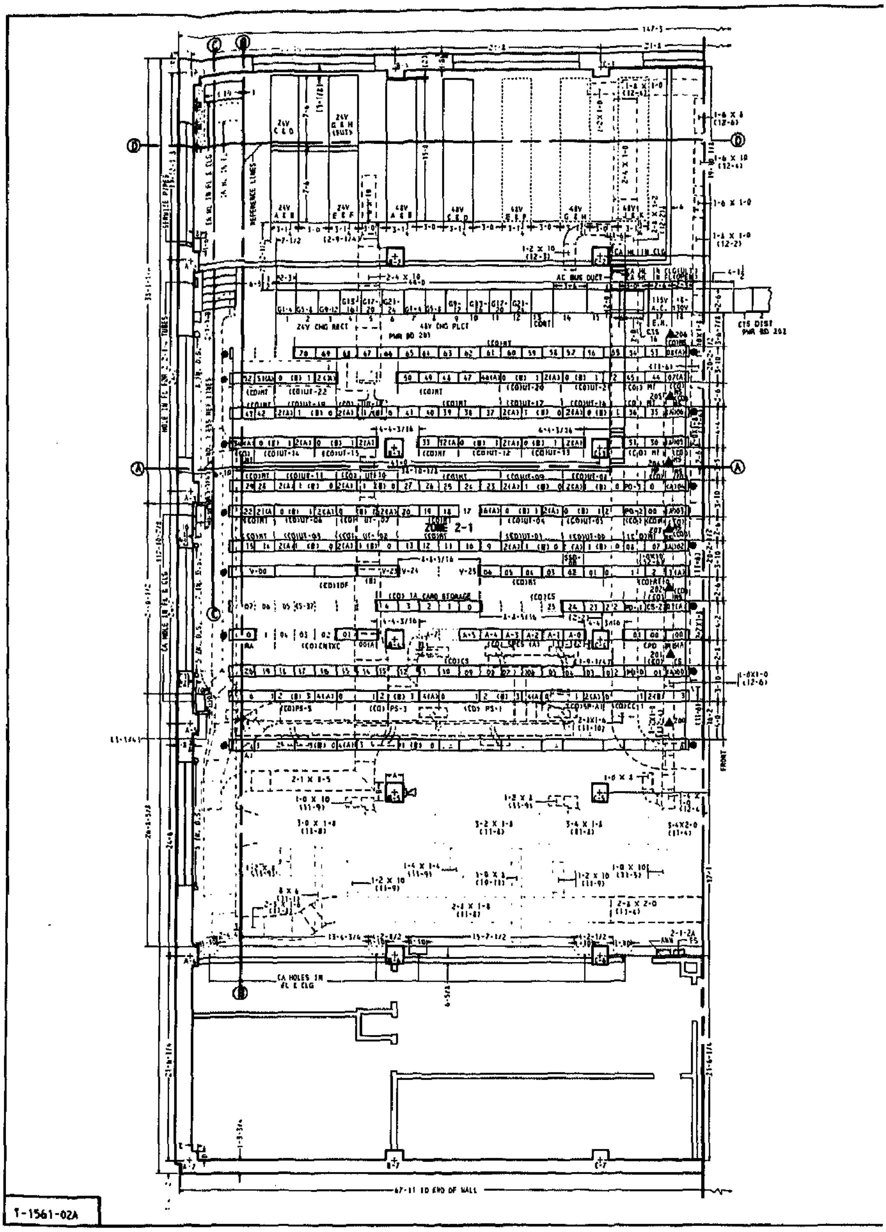

FIG. 1. A Telephone Office Floor Plan.

opment of a computer graphics system to help the draftsman produce the drawings more quickly than he can by conventional methods, but without disrupting other aspects of the telephone office engineering process.

The drawings being considered are fairly typical of a general class of architectural and engineering "blueprint" drawings. Many of their character-

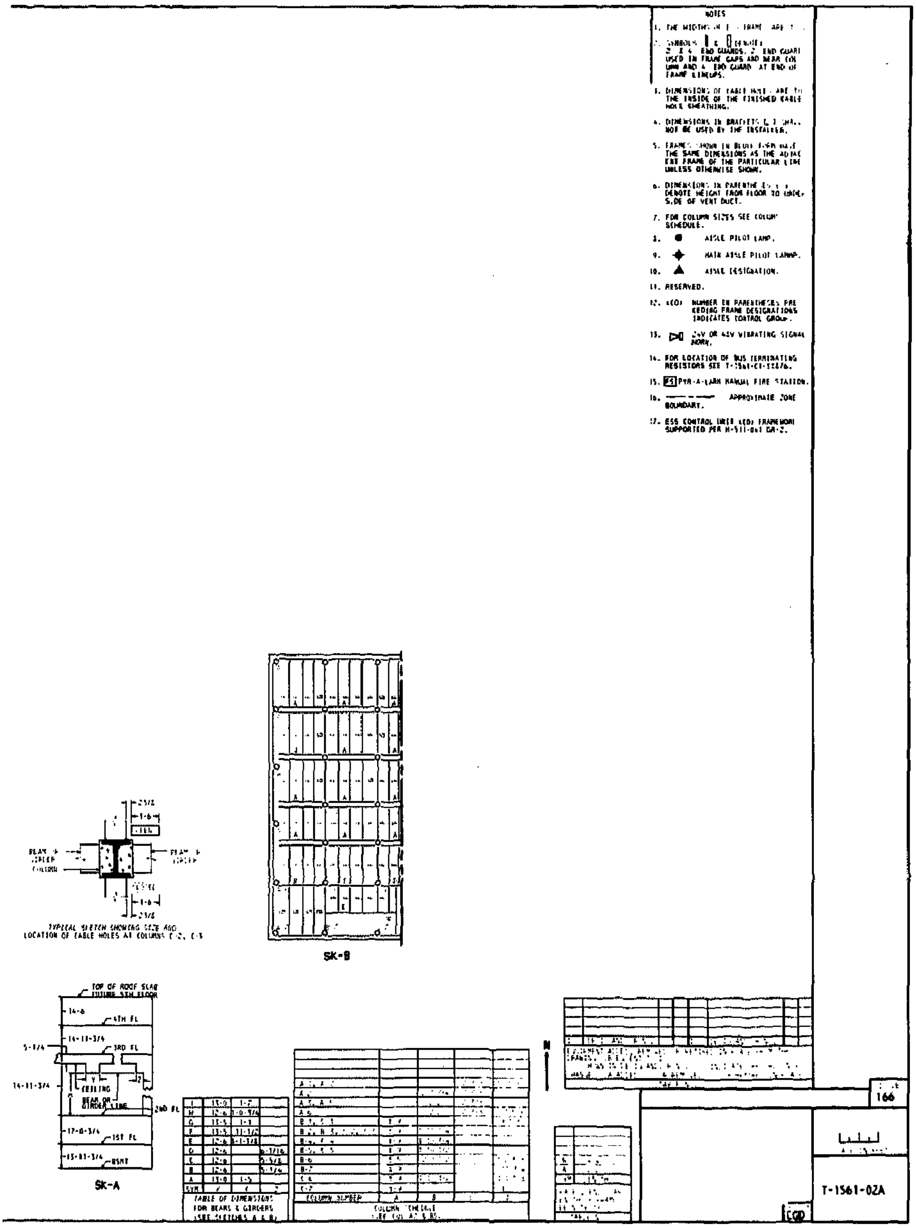

FIG. 1 (*Continued*)

istics can be inferred from the early experimental examples shown in Figs. 1 and 2. Those characteristics of the drawings which impact most heavily on the design of the system are outlined briefly below.

First, the drawings are very large (3′ × 4′) with a low percentage of large items and a correspondingly high percentage of small detail. In addition, the drawings exhibit some degree of "locality": items are not uniformly spaced

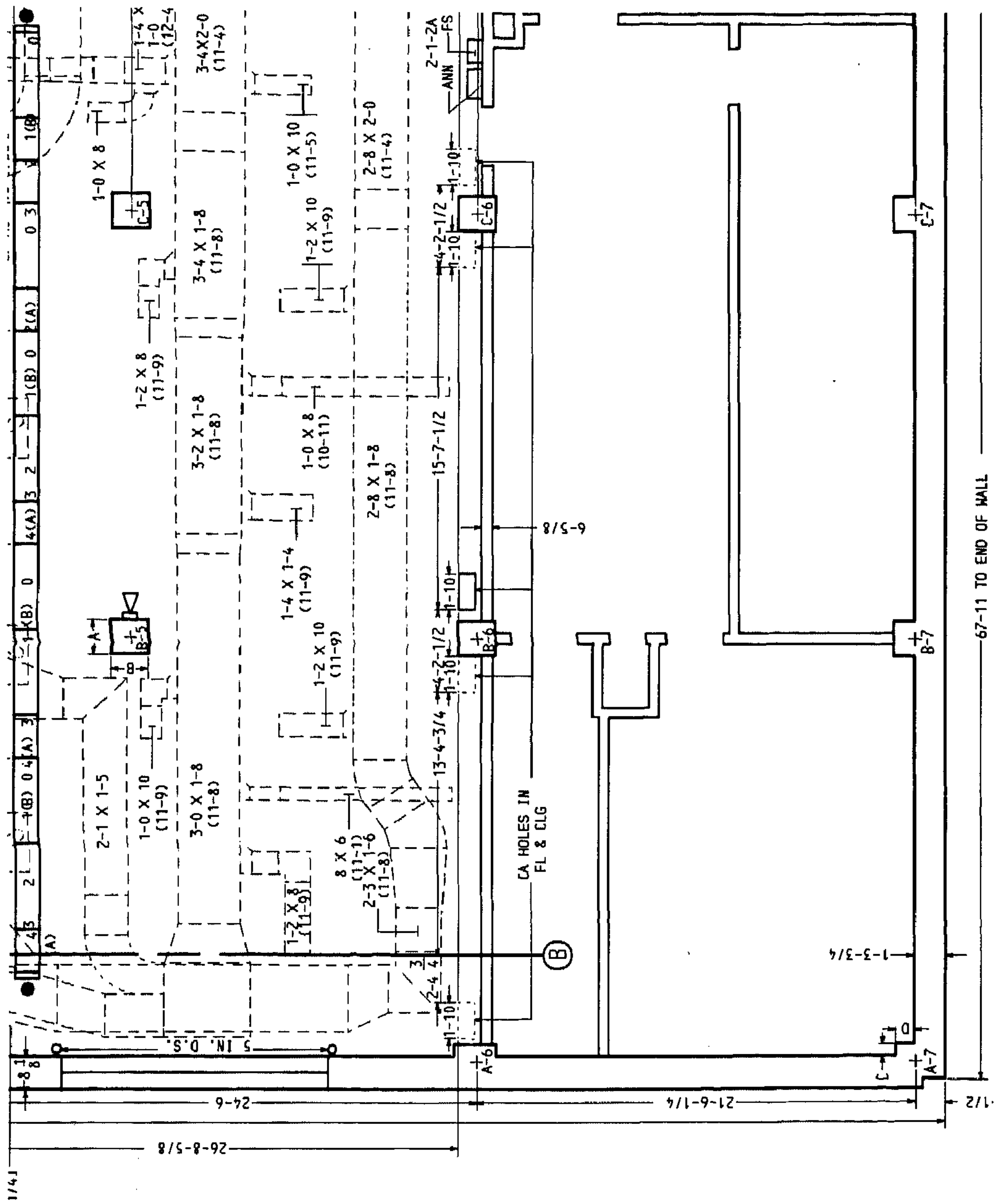

FIG. 2. Detail from Figure 1.

over the drawings, but rather tend to cluster together. This makes the drawings absolutely ideal for the application of the concept of windowing, as the draftsman's attention is nearly always focused on small detail in a small area of the drawing.

The extreme complexity of the drawings is obvious from Figs. 1, 2, and 4.

The drawings typically consist of tens of thousands of lines of varying weights and styles, and an equal amount of textual information. Not only do the drawings contain a large volume of information, but also they are made up of wide varieties of graphical entities. Further, only a small number of those possible appear on a given drawing. This aspect of the drawings makes it infeasible to rely on a "library" or cataloging technique for the creation of any high percentage of items on the drawings.

The large amount of text which is evident on the drawings poses another significant problem, especially since the text is generally comprised of one or two character labels or short words. Experience has shown that unless great care is exercised in providing flexible and general text handling facilities, a disproportionate amount of drawing time will be spent in handling text. It should, however, be noted that this is also a significant snag in manual drawing preparation: typically text is handled not by the draftsman, but by a typist on a second pass over the drawing.

The most critical aspect of the finished drawing is concerned with scaling and dimensioning. Everything on the drawing is concerned with scaling and dimensioning. Everything on the drawing must be precisely positioned according to traditional drafting dimensioning rules. This includes text, dimensioning lines, and tables. Virtually nothing is placed on the drawings arbitrarily. In addition, there are provisions for dimensioning ambiguities (arising from limited drawing precision) and arbitrary rules for resolving them.

The drawings are typically produced in families of four: for each floor plan, there is an associated lighting plan, cabling plan, and mechanical plan, with various degrees of common information among the drawings. Although this is an aspect of the drawings more or less specific to the telephone office application, it does provide an example of an area where mechanization proved to be an unarguable advantage: while the information has to be redrawn or retraced in manual production, the graphics system was designed to allow the draftsman to exploit the commonality.

Standards

The descriptions above give a general picture of the physical attributes of the drawings; the definition of a production system also requires an understanding of the logistic attributes of drawing production. The issues in this area generally revolve around one word: standards. The entire system of which the drawings are a part hinges on a rigorous set of standards which precisely define every aspect of every drawing. Their impact on the difficulty of making drawings (both manually and by computer graphics) cannot be overestimated. This phenomenon is certainly not restricted to the telephone office application, and some of the issues involved will be considered in more detail below.

First, the necessity for a complete and authoritative set of standards is unarguable. They provide the unifying thread making possible the heavy utilization of the drawings by a variety of people. They ensure that the drawings will be relatively consistent, so that they may be processed at several different levels according to well defined procedures. In addition, since they are reluctant to change, it is possible to process old drawings in relatively the same way as new ones, if necessary. For these reasons, the standards must be regarded as inviolate: a system which could not produce drawings meeting standards would be useless.

The drawing standards for telephone office engineering fill a large volume: they describe exactly how and where every line and character should be drawn; they specify the line weights and styles to be used for everything on the drawings; they specify what should be labeled and how it should be labeled; they specify what symbols should be used for any object that might possibly appear in a telephone office, from drinking fountains to cabling racks; and they cover any other imaginable aspect of the drawings. In one sense, they might be regarded as a decided advantage, for they represent (at least in theory) a very precise description of the problem to be solved in using computer graphics to do the drawings.

However, this is not the entire picture, because the extreme complexity of the standards not only leads to ambiguities and omissions, but also forecloses any hope of enforcing them all. In this sense, it is the draftsman, who interprets the standards, who is in fact regarded as the final authority. The experienced draftsman is an absolutely necessary factor in the production of useful drawings, for he makes countless decisions, based on his experience, to produce drawings consistent from both an aesthetic and an engineering point of view. For these reasons, it does not make sense to attempt to develop a system to produce drawings by replacing the draftsman; it would seem much more reasonable to view the draftsman as a resource which must be provided with better tools.

4. CHARACTERISTICS OF THE SOLUTION

The general method that was used for introducing computer graphics into the drafting process for telephone office engineering is shown in Fig. 3. The draftsman enters the drawing on the interactive graphics console, and has access to local storage (magnetic tape) for the drawing while it is in preparation. Once the drawing is completed, it is transported to a large storage file and later converted to microfilm. The microfilm is then optically enlarged to yield the final print.

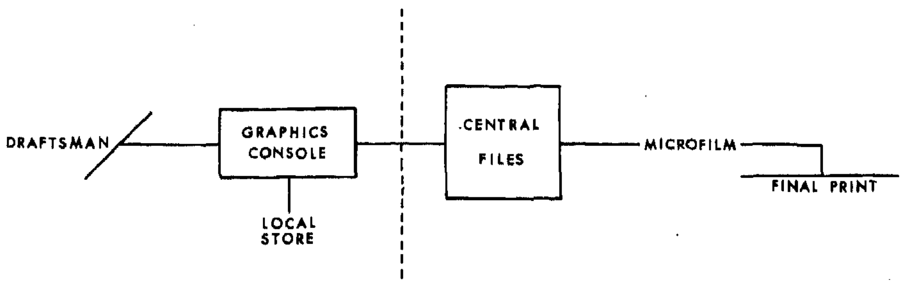

FIG. 3. General system flow.

The interfaces to existing drawing production mechanisms are smooth: the draftsman using computer graphics starts with the same sketch and drawing specifications as other draftsmen; and the computer generated microfilm is handled in exactly the same way as the microfilm made from manual drawing masters.

The creation of the drawing is of course the major step: the interactive system will therefore be described in some detail below, both in terms of the physical requirements and the functional capabilities. Also given below will be some comments on drawing storage and microfilm production procedures.

Physical Characteristics

The system is implemented in assembly language on a small (16K) general purpose computer (PDP-15) which drives a digital refresh display (GRAPHIC-2iB). Other peripheral devices used are a fixed head disk, where the picture definition is kept, and a magnetic tape unit for local picture storage and transportable output.

The display console consists of the main CRT, a 14″ × 14″ display which is the draftsman's working surface; a small storage display, which is used for reference in windowing; and the lightpen, function buttons, and alphanumeric keyboard, which are the draftsman's drawing tools.

As in most computer graphics configurations, the physical system is more precisely defined by a variety of technical parameters (precision, spot size, etc.) as well as some less technical considerations such as lighting conditions or console design. These merit serious consideration, as they have a potentially great effect on the efficiency of the draftsman who must spend hours working at the console. A system forcing the draftsman to work under limited hardware precision or tiring visual conditions would be unacceptable. Some comments on the tradeoffs involved here are given in Section 5, and a complete treatment may be found in [45]. Suffice to say at this point that the display is very high quality, with precision matching the necessary drawing precision; and that a great deal of effort went into the human engineering of the system, so that the working environment for the draftsman is comfortable and natural.

Functional Characteristics

The system allows creation and manipulation of complex pictures consisting of text and lines on a thirteen bit (8192 × 8192) grid. This provides the draftsman with a virtual display area of over 9 ft square on which he can work to a precision of 0.014″. The following functional capabilities are available for building the picture: basic creation and manipulation commands; windowing and scaling; text entry and editing; layering (overlay facilities); line style selection; library facilities; and a set of application-oriented special functions. In the descriptions below, words in capital letters will refer to the actual function names invoked (generally through lightbuttons) by the draftsman. The descriptions below will be necessarily rudimentary; they are intended only to give a general overview of the capabilities of the system and to form the context within which the more significant design and implementation considerations can be discussed.

Basic Commands

The basic line creation and manipulation commands are representative of the standard commands found in classical computer graphics draw systems. The exact mechanisms of their definition and use in the context of the drafting application are included here for the sake of completeness.

Tracking cross manipulation is a capability common to many of the functions, as it provides a method for specifying points on the drawing with the lightpen. A cross is displayed which can be moved continuously with the lightpen or incrementally with the display keyboard. The position of the cross is continuously displayed in units consistent with the drawing (generally feet and inches): the absolute position appears in the corner of the screen and the relative position (to the last defined point) moves along with the cross. Since everything on the picture is positioned through the use of the cross, there are a variety of mechanisms to allow the draftsman to manipulate it quickly and precisely. These will be covered in greater detail in the discussions on precision below.

DRAW is the general purpose line creation facility. Line endpoints are defined by the tracking cross and locked when a function button is pressed. A line can be created in "rubber band" mode, where the line endpoints are defined by the last locked point and the current position of the tracking cross, or "manhattan" (horizontal–vertical constraint) mode, where the line is defined as the horizontal or vertical line starting at the last locked point and ending at the vertical or horizontal line through the tracking cross, whichever is longer. There are many options available during DRAW, such as automatic CLOSE of the figure being drawn, and ARROWhead generation at the ends of lines.

GROUP is a subfunction which allows definition of a set of picture entities for handling by any of the manipulative functions (MOVE, COPY, DELETE, etc.). An element on the screen is SELECTed by simply lightpenning it—when the SELECT has been recognized by the system the item will blink. A blinking item can be REJECTed by entering REJECT mode and lightpenning it (it will then stop blinking). A function button is used to exit GROUP, at which time all items which are blinking are defined as the items constituting the GROUP to be manipulated.

MOVE allows the GROUPed items to be moved under tracking cross control. The tracking cross will appear on the screen after the elements have been GROUPed, and the X and Y offsets of the tracking cross from its original position will be applied to the selected items as the cross is moved.

DELETE simply erases the GROUPed items from the picture.

COPY will make a fresh copy of the GROUPed items, and then enter MOVE to allow positioning of the new copies. REPEAT is an extended copy which allows the user to define a matrix of copies of the GROUPed items. (This is often useful for repetitive work, such as drawing tables.)

SYMM allows various symmetry options to be applied to the GROUPed items. They may be complemented in X or Y, X and Y exchanged, or any combination. This allows eight possible transformations to be applied to each element: 0, 90, 180, or 270 degree rotation; or 0, 90, 180, or 270° rotation of the mirror image. The GROUPed items are rotated and reflected about a common point, so that their relative orientations are unchanged. Although it would be fairly easy to allow arbitrary rotation, there appears to be no real need for it in telephone office drawings, and this limited facility has proven sufficient.

Windowing

WINDOW is the basic command for user window manipulation. The draftsman selects any one of four window sizes, and the position of the selected window in relation to the entire picture is superimposed over his picture, along with a tracking cross. As the tracking cross is moved (and the window along with it) the picture displayed is changed to reflect the new window position. The tracking cross may be moved as fast as the user desires, but for complex pictures the display may not be updated for a few seconds (this is the "clipping time" which will be discussed below). In addition, the full picture and current window position are always displayed on the secondary storage display. This combination of information makes it very easy for the draftsman to quickly define a new window position without losing his orientation in the picture.

Text Manipulation

The draftsman enters and edits text under control of a character oriented editor, with INSERT, REPLACE, DELETE, and NEW LINE capabilities. Text may be positioned anywhere (NEW LINES are relative to the first character of a text string), and the available character sizes are preset to match standards. In addition, the text is clipped and scaled correctly (exactly as if it were composed of lines) and editing may be done at any window size and position.

Layering

The drawing can be broken up into LAYERs, which can be thought of as overlays: every item on the drawing is defined by its XY position and its layer number. Any combinations of layers can be viewed or output at any time, and one layer is designated as the active layer. Only items on the active layer may be manipulated. This capability (an idea borrowed from the integrated circuit design technology) is very useful in the telephone office application, for it allows the draftsman to create the family of four telephone office drawings with little redundancy. Everything on the drawing is defined to have a certain layer; then different combinations of layers produce the four different drawings.

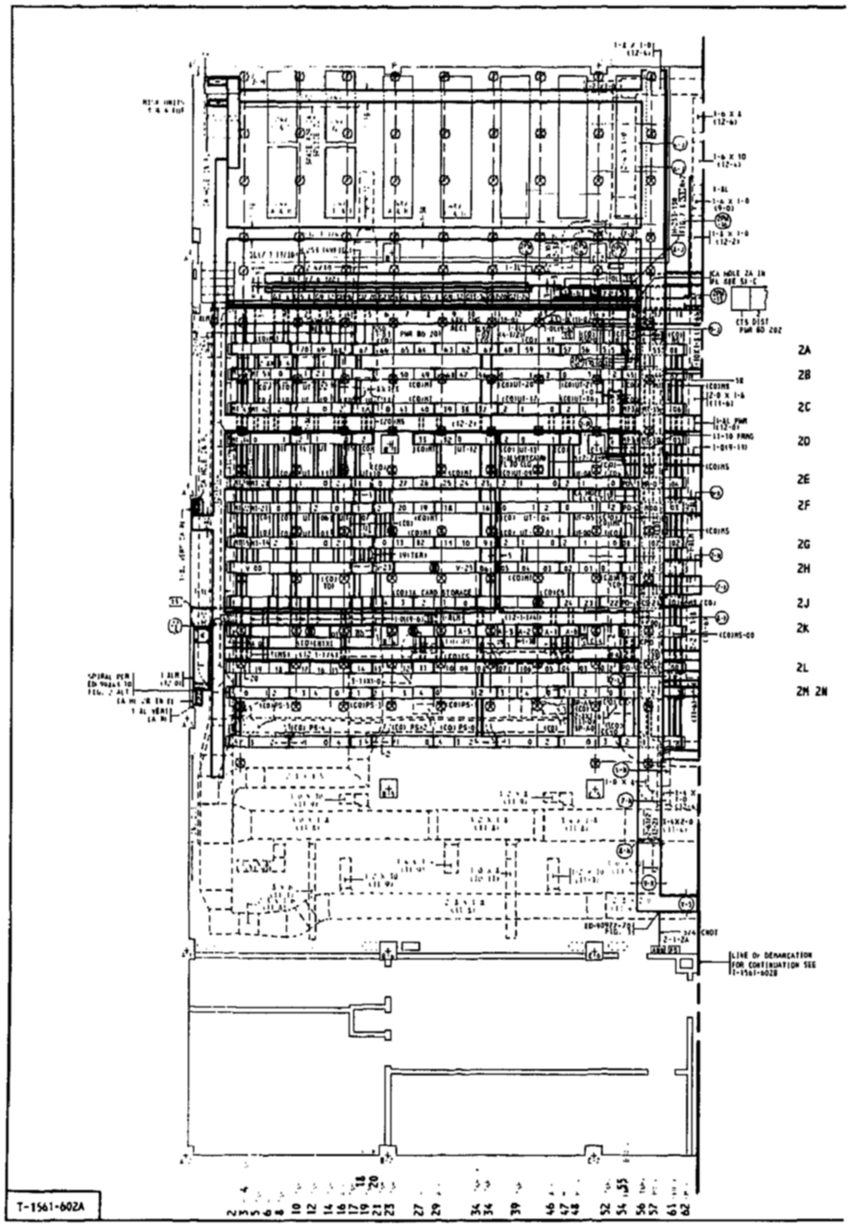

FIG. 4. Cabling plan for office in Fig. 1.

The value of this technique can be appreciated by comparing Fig. 4, a "cabling plan", with Fig. 1, the "floor plan" for the same office. There are a large number of obvious similarities between the drawings, such as: the building outline; the telephone office frames and their labels (the rows of rectangles); and the ducting (the dotted line structures most obvious at the

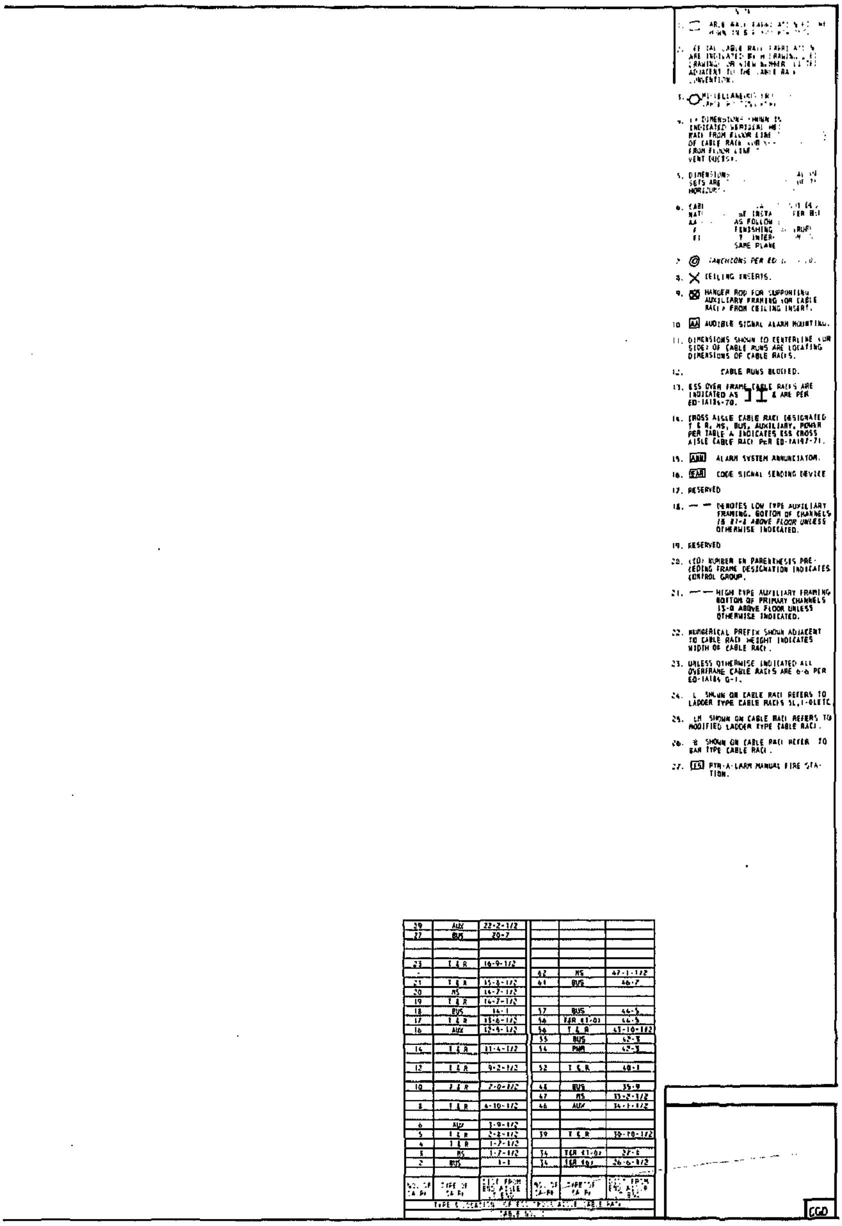

FIG. 4 (*Continued*)

bottom of the drawings). Equally apparent are the differences between the drawings: there are no dimensioning lines on the cabling plan; the building outline appears in different line styles on the two drawings; the tables and figures differ; and so on. A set of policies based on drafting standards was developed to allow this layering facility to be used to produce these drawings concurrently.

Line Styles

Every line on the drawing has one (or more) associated pen style, allowing creation of all the various line widths and dot–dash configurations specified by standards. The draftsman has a command to change the PEN style, and he does so whenever he would change physical pens on a manual drawing. Although standards specify far too many different styles for any graphics hardware to display, sixteen distinct styles are displayed. The draftsman can easily find and change the line style of any existing item. The line style is passed as a code to the microfilm generation system, which produces the exact style specified by standards.

Library

Commonly used items can be placed on a library and quickly recalled for placement anywhere on the drawing. This library is actually one of the "layers" of the drawing, and its use is therefore very flexible. In addition, there are capabilities allowing the draftsman to populate the library from a larger library on secondary storage. The usefulness of the library depends on the kinds of drawings that are being done and the effectiveness of the arrangement of the various figure libraries.

Special Functions

The final major category of capabilities of the system is comprised of a set of special purpose functions, designed to facilitate the creation of the specific kinds of drawings being done. For example, there is a function allowing the automatic creation of telephone office frames (there are hundreds of different types). Other examples are the facilities for automatically generating dimensioning lines and arrowheads, or for automatic circle and rectangle generation.

In order to illustrate the usefulness of this system for the drafting application, it is necessary to consider some of the more practical aspects of the problem. First, the commands available are essentially a complete set: any command needed by the draftsman has been included; and the system will produce an accurate finished product for drawings of arbitrary complexity. Second, the system maintains excellent response times; the only response time not instantaneous is during windowing, and even then it is limited to 5 sec for even the most complex drawings. Third, the system is implemented on a small, inexpensive computer. The interactive draw programs which have been developed in connection with design systems invariably include compromises in one of these three areas: either long response times are tolerated [27,35]; more powerful (and expensive) equipment included to minimize response time [10,11,34,39,54,59]; or restrictions made on the capabilities available. While the great savings involved generally make such compromises acceptable in design systems, this is not true for drafting. It is the combination of a complete set of capabilities, implemented on inexpensive equipment, with instantaneous response time which makes the use of computer graphics feasible for drafting.

Interfaces

Once the drawings are created, they are viewed essentially as objects for handling by a data management system. The details of these operations are not relevant to the drawing creation process, but some general comments should be considered.

First, if it is remembered that it takes on the order of 50 hr to prepare these drawings, it becomes clear that rudimentary data management will support the system for some time. The drawings are simply kept on a magnetic tape library, and the magnetic tapes are used for making microfilm, etc. Until on the order of hundreds of drawings are made, the data handling time is still very small compared to the drawing time.

After a large number of drawings have been produced, and the problems of updating and changing drawings are considered, the data management becomes the central issue. However, it is still quite separate from the drawing system, and the use of magnetic tape to interface between the two can be perfectly acceptable.

Magnetic tape is also used as input to the microfilm generation system. The data is first converted directly into the microfilm control orders to make this interface simpler. This process is more than straightforward display order conversion, and has two additional major functions: First, it must take as input one drawing consisting of many layers, and output four drawings made up of different subsets of these layers. Second, it must translate the logical line styles defined in the picture into the physical lines required by standards. This process is complicated by the fact that some lines can appear in different styles on different drawings. The end result is a microfilm copy of the drawings which (theoretically) cannot be distinguished from manual drawing masters.

Finally, it is expected that some time in the future other aspects of the telephone office engineering process may be computerized. The simple format of the drawing definition (outlined in Section 5) and the existence of an independent data base makes the drafting system amenable to this kind of extension.

5. CHARACTERISTICS OF THE DESIGN

The central issues behind the successful use of computer graphics in drafting for this application can best be understood through a careful consideration of some aspects of the design and the various tradeoffs involved in the implementation of the system. It must again be emphasized that many of these issues are classical ones in interactive computer graphics: they are considered here only within the context of the drafting application. Although this system was developed as a specific solution to a specific problem, it was always clear that a system capable of handling the kinds of drawings described above would have much more general applicability than drafting for telephone office engineering. The "bottom up" design approach was advantageous in several respects, the most important being that it allowed the introduction of a draftsman into the design process at a very early stage.

The effects of using the draftsman as designer will be explored in much more detail in the next section. Following that will be discussions of the design considerations in the four major components of the system software: the man–machine interface; the windowing processor; the data structure; and the program structure. Also included will be a short discussion on the selection of the hardware to be used for the system.

The Draftsman as Designer

It was always recognized that the system would be used as a tool for the draftsman, and the draftsman was considered an integral part of the drawing preparation process. The full ramifications of this did not become clear, however, until a basic system had been developed and a draftsman brought in to experiment with and comment on the system. It became immediately obvious that the entire man–machine interface would have to be built around the draftsman's suggestions. It was anticipated that things would have to be streamlined; for this has been the bane of interactive graphics drawing systems: it is simply easier, faster, and cheaper for a draftsman to draw a line with a ruler and pencil than with a lightpen and function buttons [75]. However, the ways in which the system was to be changed were not foreseen. It developed that to meet the draftsman's needs, it was also necessary to take advantage of his particular skills. A few examples should help to illustrate this point.

A draftsman by nature is meticulous and accurate: errors in drawing are expensive to him, and he is likely to make few mistakes during the course of a drawing. This has two effects on the design of the system. First, the command structure is built on a positive "reject" philosophy (rather than positive "accept"). That is, whenever the draftsman initiates an action, the system assumes he has done so correctly, and performs the action, while still leaving the option to reject it. (The alternate philosophy, which would be used if there is a high probability of error, would require the user to positively accept the action before proceeding to another.) Second, it is not necessary to design into the system capabilities which have the sole purpose of allowing the correction of gross errors. A perfect example of this is seen in the capability to MOVE a large number of items (say half of the picture). It might seem very natural (and even necessary) to include such a function in the drafting system. The difficulty of doing so should become apparant when the system structure is outlined later in this section. The draftsman, however, maintained that such a capability would never be used. To the draftsman, to MOVE something means to erase it and redraw it, and he could forsee only the remotest possibility that a draftsman would make an error requiring this for a very large number of items.

Another example can be seen in the effects of the data structure design on the draftsman. The classical idea of an interactive graphics system for drawing involves a very flexible data structure through which picture elements may be grouped in some type of hierarchy. This is useful in design systems, where a circuit designer, for example, could indicate a minor change in some component and have that change reflected in all occurances of that component throughout the drawing. However, this is another example of a generic class of functions which would never be used by a draftsman: such changes are obviously impossible in manual drafting. The benefits of hierarchial structuring are dubious at best when no data is to be associated with drawing elements. The draftsman values much more the complete flexibility offered by a nonhierarchial arrangement: it more closely matches the manner in which he is accustomed to working with the drawings.

The effects of these considerations on the design of the system should be obvious: the positive "reject" command philosophy makes easier the task of minimizing drawing time; the upper limit on the number of items to be manipulated simplifies the design of the manipulation commands; and the limited requirements on the data structure gives greater flexibility to the data structure manipulation routines, allowing them to be highly optimized for speed. These examples are merely representative of the effect the draftsman had on the external and internal features of the system; other examples will be apparant in the discussion of the man–machine interface in the following section.

Man–Machine Interface

The man–machine interface is comprised of two basic components: the command structure, which allows the draftsman to interact with the system; and the tracking cross manipulation and constraint mechanisms, which allow the more direct interaction between the draftsman and his picture.

The issues involved in the design of the command structure are quite straightforward, and are common to nearly every graphics application. In drafting, the major emphasis is on simplicity, as only pictorial information is being entered, and the draftsman must be able to enter it with minimal time and confusion if the system is to be effective. The command interface was therefore carefully designed so that very little prompting and button pushing would be necessary. Most tasks require five or fewer distinct actions from start to finish. For example, to COPY an item, the draftsman must: select the COPY function (lightbutton); select the items to be copied; depress (any) button to exit GROUP and get the tracking cross displayed; move the copies into position; and depress (any) button to exit the COPY function. The other functions follow virtually the same cycle: there are a limited number of system states, and the draftsman is always aware of what the system is doing. It will be noted that the reliance on the skill of the draftsman mentioned above makes the design of this type of command structure very much easier.

The second aspect of the man–machine interface is the tracking cross manipulation and constraint mechanism. An extensive amount of development went into this area because of the fact that the large majority of the drawing time is spent positioning things on the picture (this is true even in manual drafting). The positioning must be very precise, and the situation is worsened by the fact that the drawings are being digitized (even one bit errors are not tolerable).

The common solution to the precision problem is to impose an artificial grid structure on the drawing, and constrain the cross to fall only on the coarse grid points [9,27,44]. Besides the fact that this method has obvious limitations, it represents exactly the kind of artificial imposed structure that is convenient for the system designer but unacceptable to the draftsman. This type of construct must be avoided in a drafting system. The alternative, of course, is to employ accurate lightpen correlation to the items as they are defined, rather than as they are displayed, and to position the tracking cross to the full precision of the entire picture, rather than the limited precision of the current scale. Any reference to a displayed item results in positioning to its fully precise defined position rather than its possibly imprecise displayed position. The tracking cross is moved across the coarse grid defined by the current scale with the lightpen; it is positioned accurately within that grid either with the keyboard or by reference to a displayed item.

There still remains the problem of the human engineering of the cross manipulation mechanisms. Since the draftsman would be spending the majority of his time doing this, the design was based entirely on the draftsman's suggestions (and complaints) through several iterations of different human interface strategies, with the following results: The cross is defined to be in one of three states: unconstrained, axis lock, and absolute lock. In the lock states, the cross constraints are determined by the next entity pointed to by the lightpen. In axis lock state, if any horizontal (vertical) line is lightpenned, the cross is moved vertically (horizontally) to that line, and then constrained to move horizontally (vertically). Note that when the cross itself is lightpenned, this reduces to normal cross constraining. In absolute lock state, the cross is moved (precisely) to the "corner" closest to the point lightpenned on the picture. This rule is also used when items are selected for manipulation.

Although these simple rules lack the generality of the more traditional cross constraint mechanisms, they have proven powerful enough to handle virtually all situations encountered in making telephone office drawings. It is far more important that their operation is concise and natural, and that they allow the draftsman to perform many actions faster than he can by hand. The time savings in this area form a very large part of the time savings involved in the entire drawing.

Windowing

The task of windowing for large and complex pictures actually consists of two problems: the geometrical problem of deciding when a line (or a portion of a line) should be displayed; and the data organization problem of providing mechanisms insuring that all data items contributing to the current window can be accessed quickly enough for display. The first problem is the classical clipping problem, and will be considered here; the second is central to the discussion of data structures in the following section.

The main difficulty involved in implementing an efficient clipper is in effectively interfacing the known techniques with the given hardware capabilities, without compromising the integrity of the clipper. It is especially important in the drafting application to be sure that a line or character or portion thereof is displayed if and only if it falls within the current window.

The clipping of vectors is done using a software extension to the Sproull and Sutherland "out code" concept [72], and will not be presented in detail here. The basis of the algorithm is in the use of the "out codes", which provide, for each line, a compact description of the relationship between its endpoints and the window edges. These codes are used to accept and reject (without any geometric calculations) the majority of the lines, and to direct the clipping calculations when they must be performed.

Most graphics hardware will treat characters not as sets of lines, but as codes to be automatically translated into sets of lines. For this reason, in a software clipping processor, it is convenient to separate the problem of character clipping from that of vector clipping. The issues involved are very much hardware dependent, and will not be treated further.

It may be pointed out that it is not really necessary, given the current state of the technology, to do windowing and scaling by software, as it now can be done quite economically in hardware. However, even with a hardware clipper, it is still necessary to manipulate the data structure, and pass the entire picture through the clipper when the window is moved. The clipping time need only be low enough to be absorbed in the data handling time, and this was almost completely realized in the software for this system. A hardware clipper would therefore have little effect on the system performance.

Data Structure

The third major factor in the design of the system was the organization of the data structure. This is another classical problem in computer graphics, and many elegant forms of graphic data organization have been defined [28,83]. It would be beyond the scope of this paper to discuss fully the issues in this area, for it has received a great deal of attention in the literature. The general approach is to treat graphics as a kind of information retrieval problem, which is solved by hierarchically structuring the picture. This structure may be fixed [11,74,80], or a language may be defined to allow "customized" structures to be built [19,61]. In addition, when the graphic data representation problem is formulated in this way, it can fit into the context of more general systems [24,25,64,65,66]. In computer aided design systems, the problem is often expanded to allow consideration of how to efficiently model whatever is being designed [4,8,9,20].

For the drafting application, however, the only operation done on most parts of the picture is display: much of the power of the more general data structures would go unused. Also, it must be recognized that the pictures are not inherantly hierarchical to the draftsman: everything on the picture is equal in that it must be correctly positioned with respect to everything else.

Imposed hierarchies were tried, but they proved to be more of a hinderence than an aid to the draftsman. For these reasons, the decision was made to segment the picture only at the level of definition. Basically, this means that every creative command results in the definition of a new "item", and the picture is simply a collection of such "items", with no connections whatever between them.

The next design decision made was that it would not be possible to keep the pictures entirely in memory, even with the most efficient data organization. This would in fact be counterproductive to the idea of windowing, which says that only a portion of the picture will be "active" at any given time. This fact is much more obvious in retrospect, since the drawings now handled by the system typically require on the order of 100,000 words for their definition. Once this decision has been made, it is necessary to consider the design of two data structures: the "master list", or complete picture definition, which is to be kept in secondary storage (disk); and the "static display list" or current display file, which is to be kept in core.

The most costly operation to be performed on the master list is windowing, when the entire list must be accessed and passed to the clipper. In the implemented system, this is done by organizing the master list as a simple one-dimensional contiguous list of picture elements, and double buffering during the sequential scan. This was very easy to implement and performs very well, as no function requires more than one scan through the master list. However, this can be improved in one of two ways: One technique would be to leave spaces on the disk so that "seeks" during the sequential scan are mimimized. A second (probably more effective) idea would be to organize the data positionally so that the entire list need not be scanned during windowing, especially for small windows. Nevertheless, as implemented, the longest response time occurring on the system (for pictures containing tens of thousands of lines and tens of thousands of characters) is 3–4 sec; as contemplated, all response times will be less than one second.

The design of the static display list had to take into account both time and space considerations, for not only is there a limited amount of core available, but also the complexity of the static display list directly contributes to the flicker rate on the display. One design, for example, might involve keeping elements in core in exactly the same format they are kept on disk, and passing them through the clipper on each display cycle. This is in fact necessary in some cases: when an item is being moved, for example, it is necessary to bring the master list description of the item into a "transient area" in core, and to clip the whole item dynamically. The reason for this is that any part of the item has the potential for appearing on the current display at any instant. However, this is not true for most items. Nearly all of the time nearly all of the items in the window are merely being displayed, not manipulated (hence the name "static" display list). To keep them in their relatively verbose master list format in core would waste space; and to clip them on every display cycle would waste time. Therefore, the static display list is generated at the time the window position is defined, and is a highly optimized display list containing only those orders necessary to display the lines in the current window. The only piece of overhead associated with each item is a back pointer into the master list, so that the whole item can be brought into core when, for example, it is selected for manipulation.

The following description of the program flow and the example given in the following section should help to make more clear the functions of the various data structures.

Program Flow

The general flow of the program can be described in terms of the three general situations faced by the system: normal display (idle state); windowing (when the user is changing the window position); and modification (when the user is executing a command which affects the data structure).

When the system is in idle state, the picture is given to the display directly from the static display list, which was constructed by the clipper and contains the display orders defining the portions of the picture appearing in the current window. Also displayed is a menu of lightbuttons and some system status information.

If a command is invoked (a lightbutton selected), then a new menu of lightbuttons (for that command) is displayed, and the following general actions are taken. First, the user selects the items on the picture to be manipulated. These are removed from the static display list, and then read from the master list (on disk) into the transient area (in core), where they are clipped dynamically. These items are not in display order format; they are kept exactly as they appear in the master list. The clipper converts them to display orders on every display cycle. This entire process is fast enough so that it generally cannot be noticed by the draftsman. Manipulation is then done by modifications to the transient area (master list copies of the items), and changes are reflected on the display by the clipper during each display cycle. When the user is done, the master list is updated ("in-place", if possible, else "delete-and-append"), and the static display list rebuilt to reflect the changes. Commands which involve the creation of new items are handled in the same general way: the items are created in master list format in the transient area; then they are appended to the master list upon exit.

Windowing (or changing layers) involves cycling through the entire master list, and passing it through the clipper to the display. This is done through a double buffering technique, so that disk *I/O* and clipping are overlapped. In addition, the static display list is rebuilt on every cycle through the master list. (This requires no additional overhead, since the orders that go to the display are the same as those that go into the static display list.) Therefore, when the user exits windowing, the static display list is complete, and the system is once again in idle state.

In summary, consider the example detailed in Fig. 5. In Fig. 5a, the system is in idle state, displaying the circle, rectangle, and triangle within a large window. The definitions of the three are in the master list on disk, and display orders for them have been built in the static display list. The solid

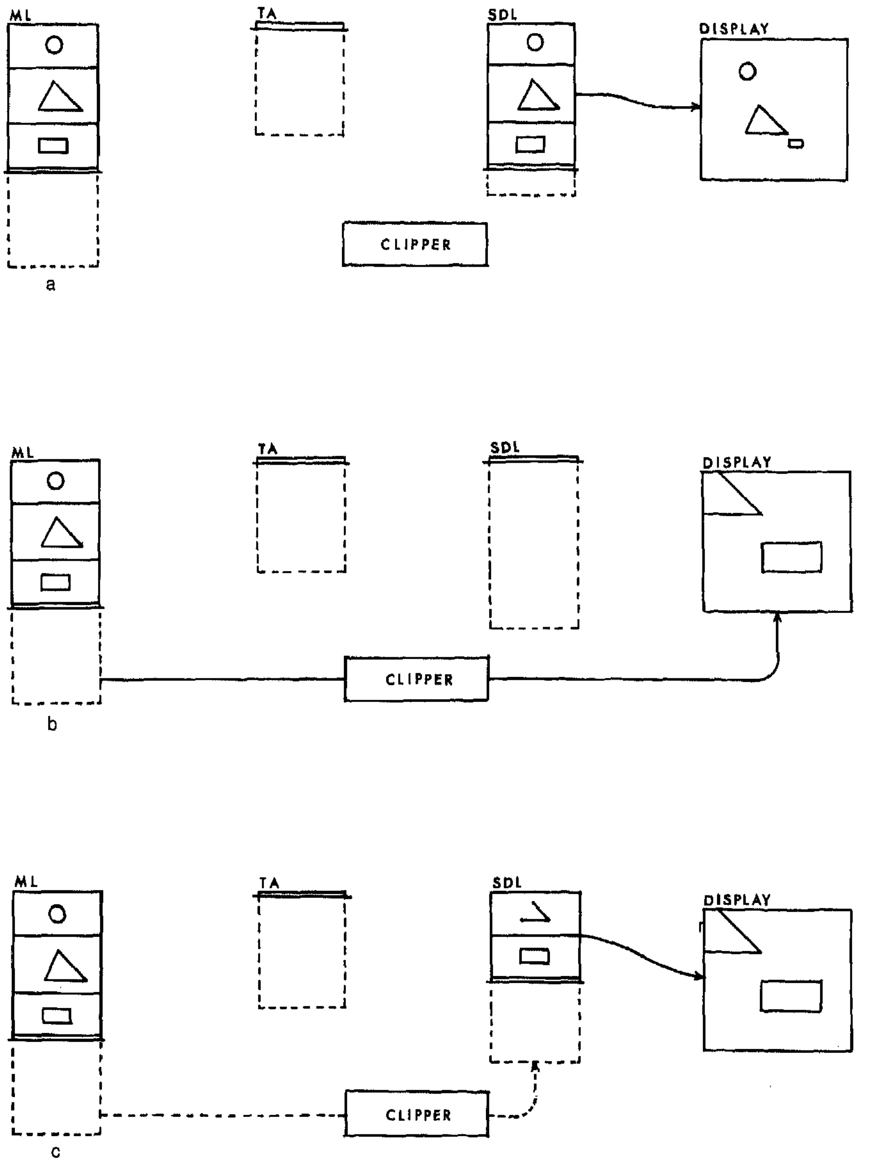

FIG. 5. (a) Idle state; (b) During windowing; (c) After windowing.

line connecting the static display list to the physical display indicates that the data in the SDL is being used to regenerate the display (dotted connecting lines will indicate one-time data transfers). Figure 5b shows the situation while the user is changing the window position (and size). The display is regenerated by passing the entire master list through the clipper. The static display list is rebuilt to contain the orders necessary to display that part of the picture in the window (in this case the rectangle and part of the triangle) as shown in Fig. 5c. In Fig. 5d, the user has selected the triangle for

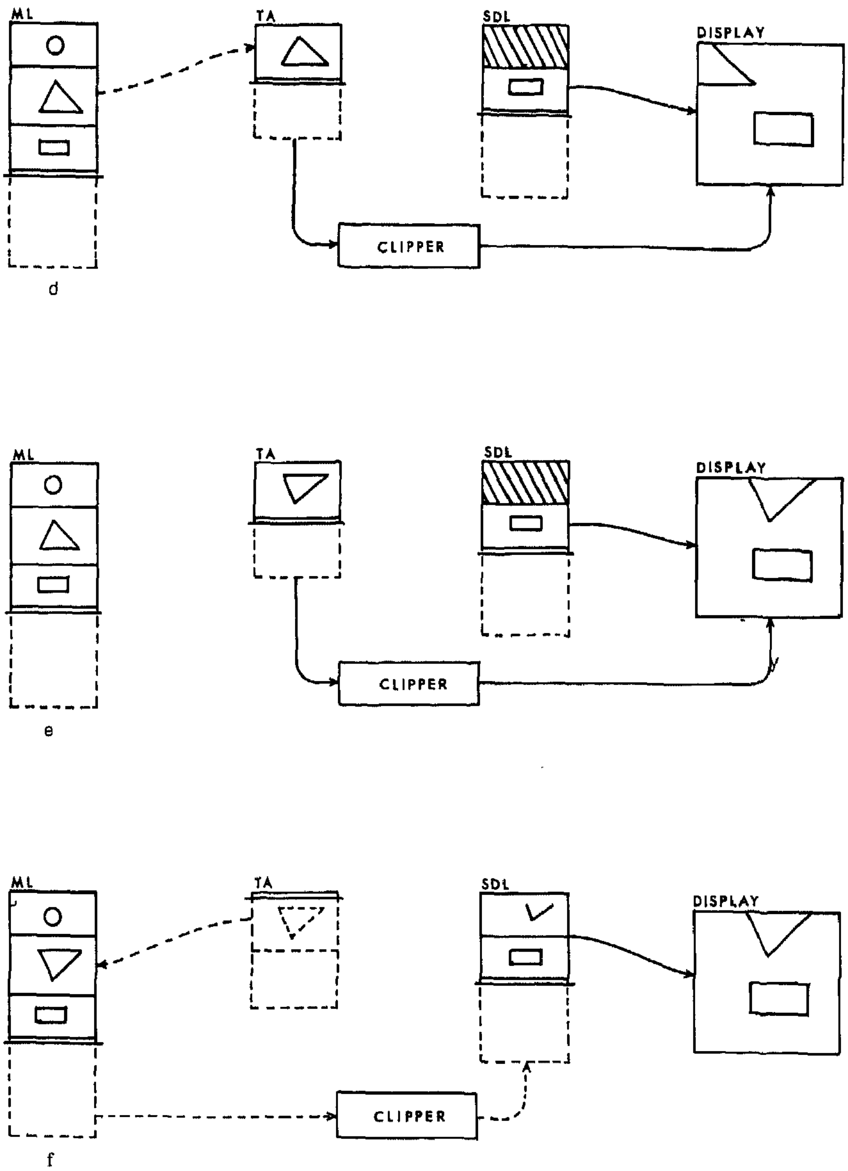

FIG. 5. (d) Begin modification; (e) During modification; (f) After modification—idle state.

manipulation, so the master list description of the triangle has been put in the transient area, and its static display list description disabled. The triangle is displayed by passing the transient area through the clipper; the rectangle is still displayed from the static display list. This allows the triangle to be clipped correctly as it is manipulated (Fig. 5e). Finally, when the manipulation is complete (Fig. 5f), the master list is updated from the transient area, and the static display list rebuilt for the new picture.

The core map for the system in operation is diagrammed in Fig. 6. The

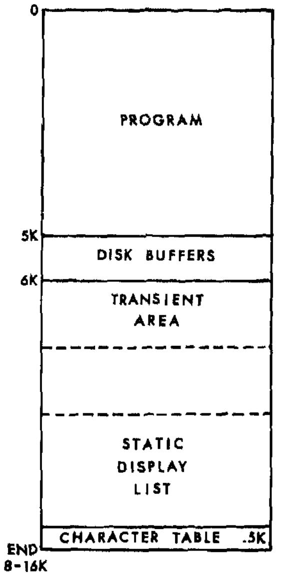

FIG. 6. System core map.

program occupies approximately 5K words, and is entirely core resident. It would be possible to overlay the various user functions, with no effect on performance, to reduce this to less than 4K words. Following the program are the disk buffers, which currently occupy 1K, but will probably be enlarged to improve performance. The last 512 words of memory contain the table used by the display character generator in forming characters. The rest of memory is used for the transient area and the static display list, which grow towards one another. The amount of core available here determines the effectiveness of the system on complex pictures. The system is completely operational on an 8K machine; in fact the drawings in Figs. 1 and 4 were done on this machine. However, the more complex the drawing and the larger the window, the more core is necessary for the static display list. It was decided worth the cost to use a 16K machine for the production system, as it ensures that core limitations will not hinder the draftsman at all in the preparation of drawings.

Hardware Selection

Interactive computer graphics has come to apply to a wide spectrum of hardware devices in various configurations. The relative merits of the various devices, however, are very application-dependent. Some of the characteristics of the hardware used in this system, as relevant to the drafting application, are given below. More complete coverage of the various issues involved in hardware selection for computer graphics may be found in [11,13,46,53,63,68,77,79,81]. Specific technical data on the present system is given in [45].

First, the flexibility offered by a refresh display is essential if a high level of interaction is to be maintained. The success of the drafting system depends on the quality of the interface between the draftsman and the machine; and the use of a less powerful interactive graphics device as the main display (e.g. digitizer–plotter or storage display) would necessitate the introduction of more artificial commands, and would weaken this interface. A digital refresh display was chosen because of the greater potential precision offered by digital displays over analog displays. The phosphor was carefully chosen to yield good flicker characteristics as well as acceptable dynamic behavior [45].

The selection of the peripheral devices to be used for interaction with the display was treated largely as a human engineering problem, and the standard lightpen, function buttons, and alphanumeric keyboard proved perfectly adequate. A well-engineered data tablet may be an acceptable substitute for the lightpen, but it was felt that the direct interaction with the picture offered by the lightpen is an advantage. The secondary storage display seems to be essential as a display peripheral, for it allows the windowing process to become very much less confusing for the user.

Finally, the problem seemed best suited to a small general purpose computer, as opposed to timesharing on a large computer or multiplexed displays on a medium size computer, for several reasons. First, the chief requirement is for best response time, and this seems best served by keeping all the data local to the console. This can be done because the actual computing requirements of the system are very low. The software is mainly concerned with maintaining the display and controlling the interaction, so that much of it would be necessary anyway in a timeshared configuration. A multiplexed display configuration was considered, but studies showed that the stand-alone configuration was more economically sound. The reason for this is that core requirements are relatively constant, and processor costs relatively low. In addition, reliability considerations and the obvious psychological advantages of keeping the system as independent and compact as possible pointed to the stand-alone processor configuration.

6. CONCLUSION

One beneficial effect of having a draftsman participate in the design process was that the transition from an experimental to a production system was very smooth. The availability of a draftsman experienced with the system and familiar with its capabilities meant that real drawings could be produced almost immediately. The system went into a production environment within Western Electric on a trial basis in early 1973 (see Fig. 7). Within the first few months, several operators were trained on the system and it went into multishift operation, with drawings being produced in the expected time range. More important, the system caught the imagination of the local drafting organization [14,18], so that it is truly viewed as a working tool rather than an experimental toy. After a full evaluation of the trial system, similar systems are planned for future installation in other locations.

Fig. 7. System in operation.

The drawings produced match standards with minor exceptions, and in many ways have a better appearance than manually produced drawings. There is every indication that the drawings will become fully integrated into the system for drawing production. It will be some time before this process will be completed, for it is being done extremely carefully, so that the large numbers of unforeseen political, logistic, and technical problems remaining may be dealt with with no disruptive effect on everyday operation.

Technically, the system has met every goal. It allows the production of telephone office drawings by interactive computer graphics while still maintaining ease of input and integrity of output. Regular professional draftsmen have quickly learned to use the system to produce drawings which meet existing standards. Further, the drafting is done with a small, inexpensive computer and display system, so that the technical and economic feasibility of drafting by computer graphics has been demonstrated.

ACKNOWLEDGEMENTS

The actual realization of this system required a great deal of effort in a variety of technical areas, most notably by Messrs. J. W. McDonald, F. S. Cilino, T. H. Aschenbrenner, and P. K. Nelson. Special thanks, however, must go to Mr. J. B. Macdonald, the driving force behind the system, and Ms. S. L. Maszk, the draftswoman who was so critical to the final stages of the project.

REFERENCES*

1. Interactive Graphics in Data Processing, Special Issue of *IBM Systems J.* 7, No. 3/4 1968. (A special issue devoted to computer graphics, covering: display concepts and data structures; language and systems support; and some specific applications. Relevant individual papers are referenced separately.)
2. S. L. Anonsen, Interactive computer graphics in architecture, *Comput. Automat.* 19, No. 8, 1970, 27–30. (A simple picture editing system developed by an architectural firm to demonstrate the feasibility of using computer graphics in architectural design.)
3. J. Atiyah and P. K. Wall, Practical layout of printed circuit boards using interactive graphics, in "Advanced Computer Graphics" (R. D. Parslow and R. E. Green, ed.), Plenum Press, New York, 1971, pp. 301–312. (A conventional system using off-line input and interactive editing for the printed circuit board layout application.)
4. H. B. Baskin and S. P. Morse, A multilevel modeling structure for interactive graphic design, *IBM Systems J.* 7, No. 3/4, 1968. 218–228. (An interactive design system is described which separates conversational (drawing) capabilities from analysis. Separate data structures are employed and translation mechanisms described between the two.)
5. C. W. Beardsley, Special applications report—computer aids for IC design, artwork mask generation, *IEEE Spectrum* 8, No. 9, 1971, 60–79. (A complete coverage of industrial use of integrated circuit mask layout systems, both "inhouse" and "turnkey".)
6. C. W. Beilfuss, Automated graphics in an industrial environment, in "Emerging Concepts in Computer Graphics" (D. Secrest and J. Nievergelt eds.), W. A. Benjamin, 1968, pp. 371–396. (A noninteractive system for creating structural steel shop drawings. Emphasis is on fitting the system to the application, and the resulting program economically produces a high volume of drafting quality output.)

* This bibliography contains several surveys and other articles and books of general relevance to computer graphics for drafting which are not explicitly referenced in the text.

7. G. BRACCHI AND D. FERRARI, A graphic language for describing and manipulating two-dimensional patterns, in "Advanced Computer Graphics" (R. D. Parslow and R. E. Green, eds.), Plenum Press, New York, 1971, pp. 263–277. (An extension to FORTRAN, designed for the integrated circuit mask generation application.)
8. G. BRACCHI AND M. SOMALVICO, An interactive software system for computer aided design: An application to circuit project, *Commun. ACM* **13**, No. 9, 1970, 537–551. (A general purpose approach to the problem of interactive computer aided design. The designer is viewed as a programmer, and is provided with two high-level languages: IMOL, a monitor language for overall control of application program execution and data definition; and COIF, a picture definition and manipulation language. The languages are interactive in that they are interpretive in nature. A common ring-oriented data structure is used by all programs in the system; structure handling primitives are provided. The presentation is abstract, and the details of display handling are not considered.)
9. J. W. BRACKETT, M. HAMMER, AND D. E. THORNHILL, "Case Study in Interactive Graphics Programming: A Circuit Drawing and Editing Program for Use with a Storage-Tube Display Terminal," MIT Project MAC TR-63, October, 1969. (A sample program for network drawing and editing, mainly focused on the problems involved in maintaining a usable network model (data structure) in correlation with graphics interaction by the user. Not intended as a production system, the program has many limitations in scope (no windowing, no use of secondary storage, etc.).)
10. S. H. CHASEN, Experience in the application of interactive graphics, in "Emerging Concepts in Computer Graphics" (D. Secrest and J. Nievergelt, eds.), University of Illinois Press, Urbana, Illinois, 1968, pp. 225–308. (An outline of the work in interactive graphics at Lockheed-Georgia. Two applications are emphasized: numerical control, which involves two-dimensional picture editing and interactive design; and aircraft design, which comprises a variety of three dimensional design techniques.)
11. C. CHRISTENSEN, AND E. N. PINSON, Multifunction graphics for a large computer system, in "Proc. 1967 AFIPS FJCC," Thompson Book, Washington, D.C., pp. 697–711. (An often referenced paper describing the design of a large scale graphical services system. Several important concepts are introduced: a wide variety of hardware devices are supported; a generalized data structure is described for device (and program) independent picture representation; a language (GRIN) is defined for manipulating the data structure and providing man-machine interaction; and the problem of connecting an "intelligent" display terminal (GRAPHIC-2) to a large host computer is considered.)
12. S. A. COONS, Computer graphics and innovative engineering design, *Datamation* **12**, 5, 1966, 32–36. (An overview of the potential applicability of interactive computer graphics for a variety of design tasks.)
13. I. W. COTTON AND F. S. GREATOREX, Data structures and techniques for remote computer graphics, in "Proc. 1968 AFIPS FJCC," Thompson Book, Washington D.C., pp. 533–544. (A general design for data and program organization in a graphics system organized on a central computer–satellite basis.)
14. T. DAU, Donna and the wizard of rolling meadows, *WE Magazine*, Western Electric, April 1973, 28–31. (The drafting system described in this paper as seen from the point of view of the users.)
15. M. L. DERTOUZOS, CIRCAL: On-line circuit design, *Proc. IEEE* **55**, No. 5, 1967, 637–654. (See Dertouzos (1972).)
16. M. L. DERTOUZOS, An introduction to on-line circuit design, *Proc. IEEE* **55**, 1967), 1961–1971. (See Dertouzos (1972).)
17. M. L. DERTOUZOS, G. P. JESSEL, AND J. R. STINGER, CIRCAL-2: General purpose one-line circuit design, *Proc. IEEE* **60**, No. 1, 1972, 39–42. (A system for interactive circuit design, with emphasis on generality and power in design and analysis techniques. Although computer graphics for picture editing is not considered, the paper is authoritative on many other aspects of interactive techniques for design.)
18. R. DUPRE, Advanced techniques in engineering graphics, Bus. Graphics, Bus. Graph **7**, No. 5, 1973, 6–9. (A very short review of computer applications in engineering graphics, with the drafting system described in this paper as example.)

19. D. Evans and A. Van Dam, Data structure programming system, in "Proceedings 1968 IFIP Congress" (A. J. H. Morrell, ed.), North-Holland, Amsterdam, 557–564. (A language, information retreival facility, and paging mechanism are described which give the capability for defining and using arbitrarily complex data structures at a low level. The view is expressed that, for computer graphics, the search for a useful general purpose data structure is fruitless. The system described provides an alternative, as it allows applications programmers to build "customized" data structures.)
20. D. S. Evans and J. Katsenelson, Data structure and man-machine communication for network problems, *Proc. IEEE* **55**, No. 7, 1967, 1135–1144. (A system for electronic circuit analysis which emphasizes the need for appropriate data structure design. The user essentially interacts with the data structure modelling his network; computer graphics is used to help facilitate this.)
21. D. K. Ewing and D. W. Tedd, A display system for processing engineering drawings, in "Advanced Computer Graphics" (R. D. Parslow and R. E. Green, eds.), Plenum Press, New York, 1971, pp. 245–259. (An experimental system giving specific approaches to some problems involved in editing three dimensional drawings: chiefly data structures, picture transformations, and lightpen correlation. The argument is made that graphics is useful only for rough editing as opposed to precision input.)
22. M. Faiman and J. Nievergelt, (eds.), "Pertinent Concepts in Computer Graphics," University of Illinois Press, Urbana Illinois, 1969. (Proceedings of the 1968 Conference on Pertinent Concepts in Computer Graphics at the University of Illinois. Relevant individual papers are referenced seperately.)
23. G. R. Fair, A. D. J. Flowerdew, N. G. Munro, and P. Rowley, Note on the computer as an aid to the architect, *Comput. J.* 9, No. 1, 1966, 16–20. (A very ad hoc system showing the potential for having the computer do routine calculations on simple architectural designs. See Newman (1966).)
24. J. A. Feldman, "Aspects of Associative Processing," MIT Lincoln Laboratory Technical Note No. 1965–13, April, 1965. (A discussion of the early work on associative processing which led to the development of the LEAP language. See Feldman and Rovner (1969).)
25. J. A. Feldman and P. D. Rovner, An Algol-based associative language, *Commun. ACM.* **12**, No. 8, 1969, 439–449. (A discussion of LEAP, a high level language for associative processing, and a software implementation of it using hash-coding techniques. The language and concepts developed are of very general scope: they are relevant to computer graphics in that they represent a major alternative to the typical master-instance ring structure for picture representation. See Feldman (1965), Rovner and Feldman (1968), and Newman and Sproull (1973).)
26. B. W. Fowler, Xymask, A pattern generation language, *Bell Lab. Rec.* 47, No. 6, 1969, 205–219. (The picture description language used by Bell Laboratories for integrated circuit mask processing.)
27. L. J. French and A. H. Teger, GOLD—A graphical on-line design system, "Proc. 1972 AFIPS SJCC," AFIPS Press, Montvale, N.J., pp. 461–470. (A system which uses classical techniques for creating and editing integrated circuit artwork on a refresh display driven by a small computer. A hierarchical data structure is used and a variety of interactive picture editing facilities provided. The paper also includes an analysis of the factors making up picture compilation time.)
28. J. C. Gray, Compound data structure for computer-aided design: a survey, in "Proceedings ACM 22nd National Conference 1967," MDI Publications, Wayne, Pa., pp. 355–365. (A coverage of the various data structures which have been invented for use as models in computer aided design problems. Described are: Sketchpad ring structures; CORAL; APL; ASP; L6; AED; and AL (precursor of LEAP).)
29. T. G. Hagan and R. H. Stotz, The future of computer graphics, "Proc. AFIPS 1972 SJCC," AFIPS Press, Montvale, N. J., pp. 447–452. (A coverage of future prospects for computer graphics, with emphasis on varied application areas. Conclusion: "In interactive computer graphics, we are at the end of the beginning.")
30. H. R. Harris and O. D. Smith, Autodraft—A Language and processor for design and drafting, in "Proceedings 1965 SHARE Design Automation Workshop." (A picture de-

scription language and processing system at North American Aviation specifically designed for the production and handling of high quality aerospace engineering drawings.)

31. L. HAZLETT, Computer graphics for integrated circuit design, in "Advanced Computer Graphics" (R. D. Parslow and R. E. Green eds.), Plenum Press, New York, 1971, pp. 313–324. (A good description of a complete production system for IC mask design and generation. The paper describes the entire process, but gives few details on software implementation or performance.)
32. P. G. HENRY AND D. H. MOUNTFORD, Production of pipeline isometric drawings using a digital computer and an incremental plotter, in "Proceedings 1968 IFIP Congress" (A. J. H. Morrell, ed.), North-Holland, Amsterdam, pp. 1422–1426. (A noninteractive ad hoc system designed to produce dimensioned center-line pipeline isometric drawings on a plotter.)
33. E. L. JACKS, A laboratory for the study of graphical man-machine communication, in "Proc. 1964 AFIPS FJCC," Spartan Books, Baltimore Md., pp. 343–350. (A description (along with four related papers in the some volume) of the General Motors DAC-I system, the first large-scale effort of applying computer graphics to industrial design problems.)
34. E. L. JACKS, Observations on a graphic console system, in "Emerging Concepts in Computer Graphics" (D. Secrest and J. Nievergelt, eds.), W. A. Benjamin, 1968, 131–170. (An abstract assessment of the overall value of the General Motors DAC-I system in engineering design. See Jacks (1964).)
35. J. F. JARVIS, The design of interactive graphics aids to mask layout, *Proc. IEEE* **60**, No. 1, 1972, pp. 35–39. (A short general review of the problems involved in using interactive graphics for integrated circuit mask layout, followed by a description of a particular system (IMP). The IMP program is basically a picture editing system implemented on a small stand alone terminal. It is very flexible and includes a full set of capabilities, but response times of one minute and more are tolerated.)
36. C. I. JOHNSON, Principles of interactive systems, *IBM Systems J.* **7**, No. 3/4, 1968, 147–173. (A survey article with an extensive bibliography detailing the principal programming aspects of research activity in interactive computer graphics. Discussed are: systems concepts; data structures; high-level languages; and image generation techniques.)
37. T. E. JOHNSON, Sketchpad III, a computer program for drawing in three dimensions, in "Proc. AFIPS 1963 SJCC," Spartan Books, Baltimore, pp. 347–353. (An extension to the Sutherland Sketchpad system to include a three dimensional wire frame drawing capability.)
38. A. C. KILGOUR, Computer graphics applied to computer-aided design, *Comput. Bull.* **15**, No. 7, 1971, 18–23. (Some specific applications of standard interactive techniques are given in a variety of problem areas, including circuit analysis and printed circuit board layout.)
39. J. J. LAVICK, Computer aided design at McDonnel Douglas, in "Advanced Computer Graphics" (R. D. Parslow and R. E. Green, eds.), Plenum Press, New York, 1971, pp. 195–217. (An overview of interactive computer graphics activity in an organization apparantly committed to productive use of computer graphics for design. An extensive amount of software is considered, in application areas such as Aerodynamics, Avionics, Structural Analysis and Optimization, and Numerical Control. A complex drafting package is also described, as a means by which the designer can construct a model (data structure) on which subsequent analysis is to be based.)
40. J. C. R. LICKLIDER, On-line man-computer communication, in "Proc. 1962 AFIPS SJCC," National Press, Palo Alto, Ca., pp. 113–125. (A very early survey article on potential applications of man-computer communications systems.)
41. N. LINDGREN, Human factors in engineering, *IEEE Spectrum* **3**, No. 4, 1966, 62–72. (A general survey article on human factors which includes some comments on the man–machine interface in computer graphics.)
42. A. C. LOW, Drawings, in and out, in "The Use of Computers in Engineering Design" (T. T. Furman, ed.), The English Universities Press, London, 1970, pp. 98–111. (A practical description of the various kinds "of drafting machines" used in making engineering drawings.)

43. W. J. LUZADDER, "Basic Graphics," Prentice-Hall, Englewood Cliffs N.J., 1968. (A comprehensive text on engineering drafting, which contains some comment on the automation of the drafting process.)
44. D. K. LYNN, Computer-aided layout system for integrated circuits, *IEEE Trans. Circuit Theory* **CT-18**, No. 1, 1971, 128–138. (Interactive graphics used as one of a variety of inputs to an integrated circuit placement and layout system. The system as a whole is quite flexible, and the graphical editing (which is fairly conventional) is used only when automatic placement is not practical.)
45. J. B. MACDONALD, A refresh display drafting system, in "Proc. 1973 Computer Aided Drafting Conference," IPC Science and Technology Press, 1973. (An outline of some of the problems involved in applying computer graphics to drafting, together with a brief description of the system described in this paper.)
46. C. MACHOVER, Computer graphics terminals—a backward look, in "Proc. AFIPS 1972 SJCC," AFIPS Press, Montvale, N.J., pp. 439–446. (A general overview of developments in graphics hardware and applications during the period 1967–1972.)
47. M. A. MACLEAN AND H. G. BROWN, Interactive graphical input for circuit analysis, in "Advanced Computer Graphics" (R. D. Parslow and R. E. Green, eds.), Plenum Press, New York, 1971, 279–300. (A fairly conventional system allowing creation and analysis of small circuits from standard components.)
48. J. MADDEN, Computer aided draughting of piping isometrics, "The Use of Computers in Engineering Design" (T. T. Furman, ed.), The English Universities Press, London, 1970, pp. 111–139. (A noninteractive system for producing quality pipeline isometric drawings.)
49. M. D. MANCUSI AND J. C. WILD, Interactive graphics for schematic editing, a working tool, in "Proceedings Of The ACM-IEEE Design Automation Workshop, Dallas, Texas, June 1972," pp. 301–304. (A picture editing system for logic drawings implemented on a stand-alone display system. The drawings are not excessively complex (no windowing is used; text is entered but not displayed) and many aspects of the system are oriented towards a specific kind of drawing. The program is in production use, but is just competetive with manual methods.)
50. C. H. MAYS, A brief survey of computer-aided integrated circuit layout, *IEEE Trans. Circuit Theory* **CT-18**, No. 1, 1971, 10–13. (As described by the title. Interesting because for this application, interactive graphics techniques are classed as "manual".)
51. W. M. NEWMAN, An experimental program for architectural design, *Comput. J.* **9**, No. 1 1966, 21–26. (A program which allows creation of architectural designs from a number of standard industrialized units (walls, windows, doors, etc.). A simple windowing facility based on a coarse grid structure is used to do large drawings. Some calculations (such as area finding) can be performed on the designs. See Fair (1966).)
52. W. M. NEWMAN, A prototype low cost single user graphic system, in "Graphic Languages" (F. Nake and A. Rosenfeld, ed.), North-Holland, Amsterdam, 1972, pp. 291–301. (A simple, elegant high-level language (LOGO) for defining pictures on a small single user graphic system.)
53. W. M. NEWMAN AND R. SPROULL, "Principles of Interactive Computer Graphics," McGraw-Hill, 1973. (The long-awaited first basic text on computer graphics. A comprehensive coverage of the fundamental concepts, from hardware characteristics to hidden line algorithms and advanced data structure techniques.)
54. R. E. NOTESTINE, Graphics and computer-aided design in aerospace, "Proc. AFIPS 1973 NCC," AFIPS Press, Montvale, N. J. pp. 629–633. (A description of the pragmatic and economic considerations involved in installing a large-scale graphics system (CADAM) for use in a variety of application areas involving design, drafting, lofting, and numerical control.)
55. M. G. NOTLEY, A graphical picture drawing language, *Comput. Bull.* **14**, No. 3, 1970. (A description of an elegant language allowing conversational construction of three dimensional pictures on a DVST.)
56. R. D. PARSLOW AND R. E. GREEN (eds.). "Advanced Computer Graphics: Economics, Techniques and Applications," Plenum Press, London, 1971. (Proceedings of the Computer

Graphics 70 International Symposium at Brunel University. Relevant individual papers from this volume are referenced separately.)

57. R. D. PARSLOW, R. W. PROWSE, AND R. E. GREEN (eds.). "Computer Graphics: Techniques and Applications," Plenum Press, London, 1969. (Proceedings of the 1968 Brunel University International Computer Graphics Symposium. Several survey articles are included, as well as a number of short papers on applications (mostly in Great Britain).)
58. M. D. PRINCE, Man-computer graphics for computer-aided design, *Proc. IEEE* **54**, No. 12, 1966, 1698–1708. (A general survey paper, with specific examples from the work at Lockheed-Georgia. See also Prince (1971).)
59. M. D. PRINCE, "Interactive Graphics for Computer-aided Design," Addison–Wesley, Reading, Mass., 1971. (A complete coverage of the experience at Lockheed in applying interactive graphics to aircraft design at the production level. A variety of examples are given, and economic as well as technical effectiveness is considered. Many basic principles in interactive computer aided design are developed. The introduction by I. Sutherland is also of interest.)
60. F. K. RICHARDSON AND D. R. OESTREICHER, Computer assisted integrated circuit photomask layout, in "Pertinent Concepts in Computer Graphics" (M. Faiman and J. Nievergelt, eds.), University of Illinois Press, 1969, pp. 344–355. (An early mask making system, implemented in LEAP on the TX-2 at MIT, which uses interactive graphics in building comparatively simple patterns. Of particular interest is the command interface: the user draws simple symbols freehand on a tablet to invoke commands.)
61. L. G. ROBERTS, Graphical communication and control languages, in "Proceedings of the Second Congress on the Systems Sciences, 1964," pp. 211–217. (An outline of CORAL, a graphics language based on efficient manipulation of multidimensional ring structures. The language was implemented on the MIT TX-2, and many of the concepts within it are based on the Sketchpad experience.)
62. R. L. ROSENFELD, Computer assisted mask production, *Proc. IEEE* **57**, No. 9, 1969, 1629–1634. (A noninteractive graphical description language approach to integrated circuit mask making.)
63. C. W. ROSENTHAL, Increasing capabilities in interactive computer graphics terminals, in "Proceedings of the ACM-IEEE Design Automation Workshop, Dallas, Texas, June, 1972." pp. 317–323. (A short review of interactive graphics as seen from Bell Laboratories, oriented towards terminal capabilities as relevant to the integrated circuit, printed circuit board, and electrical schematic applications.)
64. D. T. ROSS, The AED approach to generalized computer-aided design, in "Proceedings ACM 22nd National Conference 1967," MDI Publications, Wayne, Pa., pp. 367–385. (An effort of extremely broad scope in the area of man-machine problem solving. The concepts involved are probably more related to the field of translator writing systems, but the approach, in its generality, encompasses some aspects of computer graphics..
65. D. T. ROSS AND J. E. RODRIGUEZ, Theoretical foundations for the computer-aided design system, in "Proc. 1963 AFIPS SJCC," Spartan Books, Baltimore, pp. 305–322. (see Ross (1967).)
66. P. D. ROVNER AND J. A. FELDMAN, The LEAP language and data structure, in "Proceedings 1968 IFIP Congress" (A. J. H. Morrell, ed.), North-Holland, Amsterdam, pp. 579–585. (See Feldman and Rovner (1969).)
67. D. SECREST AND J. NIEVERGELT (eds.). "Emerging Concepts in Computer Graphics," W. A. Benjamin, 1968. (Proceedings of the 1967 Conference on Emerging Concepts in Computer Graphics at the University of Illinois. Relevant individual papers are referenced separately.)
68. R. A. SIDERS, *et al.*, "Computer Graphics: A Revolution in Design," American Management Association, New York, 1966. (A complete study of the "power and potential" of computer graphics in industrial applications. The economic, technological, and sociological aspects of the subject are fully covered, and a variety of specific systems are analyzed and placed in perspective.)
69. P. SMITH AND B. T. TORSON, Some comments on supporting multiple interactive graphic consoles in a productive environment, in "Advanced Computer Graphics" (R. D. Parslow

and R. E. Green, eds.), Plenum Press, New York, 1971, pp. 41–47. (*A discussion of computer graphics at Rolls Royce. Economic considerations and some applications are treated in a general manner. Most of the work is oriented towards design, and one conclusion is that drafting per se is only economically viable if most of the defined geometry is defined automatically or can be used more than once.*)

70. H. C. SO, OLCA, An on-line circuit analysis system, *Proc. IEEE* **55**, No. 11, 1967, 1954–1961. (An early system using interactive graphics as input to a circuit analysis program.)

71. A. SPITALNY AND M. J. GOLDBERG, On-line graphics applied to layout design of integrated circuits, *Proc. IEEE* **55**, No. 11, 1967, 1982–1988. (A system using limited interactive graphics (input on cards, editing by graphics) to provide input to a placement and routing algorithm for integrated circuit masks.)

72. R. F. SPROULL AND I. E. SUTHERLAND, A clipping divider, in "Proc. 1968 AFIPS FJCC," Thompson Book, Washington D. C., pp. 765–776. (Description of a fast algorithm and a hardware device to perform clipping.)

73. C. M. STRAUSS AND S. POLEY, A three dimensional piping design program, in "Proceedings 1968 IFIP Congress" (A. J. H. Morrell, ed.), North-Holland, Amsterdam, pp. 1431–1440. (A complete system using interactive graphics for the specific application of piping design and flexibility analysis. The system allows interactive definition and editing of piping networks with a variety of full three dimensional viewing modes (including stereo). The user interface and some technical aspects of the analysis algorithms are described in some detail.)

74. I. E. SUTHERLAND, "Sketchpad: A Man-Machine Graphical Communication System," MIT Lincoln Laboratory Technical Report No. 296, Jan., 1963. A version appears in "Proc. 1963 AFIPS SJCC," Spartan Books, Baltimore, pp. 329–346. (A thorough consideration of the design and implementation of the father of all interactive computer graphics systems. Though nearly all of the work is pioneering, the most notable aspects of the system are the flexible and powerful constraint definition and satisfaction mechanisms and the ring structure for picture representation.)

75. I. E. SUTHERLAND, Computer graphics: ten unsolved problems, *Datamation* **12**, No. 5, 1966, 22–26. (A brief outline of some areas of interest in computer graphics research. A problem mentioned which is of obvious relevance to drafting is the necessity for providing a smooth man-machine interface: "A draftsman with a choice of using a computer system or his old familiar pencil and paper will choose to use the computer only if it actually makes his job easier.")

76. I. E. SUTHERLAND, Computer displays, Scientific American, September, 1966. (A classic and inspirational survey article covering all aspects of the field.)

77. W. R. SUTHERLAND, J. W. FORGIE, AND M. V. MORELLO, Graphics in time sharing: a summary of the TX-2 experience, in "Proc. 1969 AFIPS SJCC," AFIPS Press, Montvale, N. J., pp. 629–642. (An overview of the organization and use of the TX-2 system at MIT, on which Sketchpad and LEAP, among other systems, were developed.)

78. B. T. TORSON, Interactive computer graphics at Rolls-Royce, in "Pertinent Concepts in Computer Graphics" (M. Faiman and J. Nievergelt Eds.). University of Illinois Press, Urbana Illinois, 1969, pp. 356–364. (See Smith and Torson (1971).)

79. A. VAN DAM, Computer driven displays and their use in man–machine interaction, in "Advances in Computers" (F. L. Alt and M. Rubinoff, eds.). Academic Press, New York, 1966, pp. 239–290. (A survey and tutorial article which describes in detail principles of display operation and performance as well as the concepts involved in employing displays for man-machine interaction. A full bibliography is included.)

80. A. VAN DAM AND D. EVANS, A compact data structure for storing, retrieving, and manipulating line drawings, in "Proc. 1967 AFIPS SJCC," AFIPS Press, Montvale, N. J., pp. 601–610. (A master-instance picture representation and specification scheme (PENCIL) which gains economy and power by virtue of the fact that it is embedded in a more general on-line system (MULTILANG) for information storage and retrieval.)

81. A. VAN DAN AND J. C. MICHENER, Storage tube graphics–a comparison of terminals, in "Advanced Computer Graphics" (R. D. Parslow and R. E. Green, eds.), Plenum Press, New

York, 1971, pp. 851–886. (A complete survey of the capabilities and characteristics of direct view storage tube terminals, which, because of their low cost, are found in many computer graphics systems.)

82. R. L. WEXELBLAT AND H. C. FREEDMAN, The MULTILANG on-line programming system, in "Proc. 1967 AFIPS SJCC," AFIPS Press, Montvale, N.J., pp. 559–569. (See van Dam (1967) for the application of this system to computer graphics.)

83. R. WILLIAMS, A survey of data structures for computer graphics systems, *Comput. Surveys* 3, No. 1, 1971, 1–22. (General comments on the need for data structures and a description of the basic kinds of organization are followed by specific representative examples of languages for creating and manipulating data structures. In addition, some typical graphics systems and data structures which have been used on them are described. An extensive bibliography is included.)

84. J. S. ZUCKER AND B. J. CRAWFORD, Graphical layout system for integrated circuit mask design, *IEEE Trans. Circuit Theory* **CT-18**, No. 1, 1971, 163–173. (A complete system for building integrated circuit masks on a small stand-alone display system. An elaborate grid system and windowing-scaling capabilities, along with a full complement of interactive commands (in addition to placement routines and library maintenance programs), make the design of complex masks possible. No details are given on software implementation or performance.)