Application Program

H20-0078-1

1620 Drafting System (1620-CX-04X)

Application Description

This program can be used by any industry to produce a drawing as computer output. The program is designed particularly to produce mechanical detail drawings from language statement input.

This manual contains a general description of the application area, application approach, general systems chart, machine configuration, and a sample problem.

Copies of this and other IBM publications can be obtained through IBM branch offices. Address comments concerning the contents of this publication to IBM, Technical Publications Department, 112 East Post Road, White Plains, N.Y. 10601

CONTENTS

INTRODUCTION

The IBM 1620 Drafting System produces detailed engineering drawings on the IBM 1627 Plotter from English-like statements. The system includes a language to describe a drawing and a set of programs which convert the language statements to a drawing.

The language consists of geometric statements which define object lines; arithmetic statements which calculate numeric values; drawing statements which add notes and dimensions to a drawing; and control statements which perform functions, such as the storage of user-defined subdrawings.

The programs of the 1620 Drafting System form a multiphase processor that is integrated with the operation of the IBM 1620 Monitor. All the programs and data of the system reside permanently in disk storage.

This is the first IBM program designed to produce detailed engineering drawings and is an important step in the installation of a Management Information System. The general graphic capabilities of the system make it applicable to a wide range of drawing requirements.

THE DRAFTING LANGUAGE

Figure 1 is a drawing produced with the 1620 Drafting System. The system includes the language used to describe a drawing and the computer processor used to translate the language statements into a drawing on the IBM 1627 Plotter. There are four kinds of statements in the language: geometric, arithmetic, drawing, and control statements.

Geometric Statements

Geometric statements define points, lines, circles, and arcs. Definitions may be written in terms of coordinates or as functions of other geometric definitions.

For example, a point can be defined by the statement:

```
P1 = POINT/2, 3
```

In this statement P1 is an arbitrary name assigned to the point, the language word POINT indicates that a point is being defined, and the numbers 2 and 3 are the coordinates of the desired point.

A horizontal line three inches long is defined by the statement:

```
LINE/ DX, 3
```

In this example the word LINE indicates that a line is being defined, the language word DX means horizontal, and the 3 shows the desired length.

One way to define a circle is:

```
CIRCLE/ 3, 2, 2
```

The center of the circle is the point (3, 2), and the radius is two inches.

The language also has statements which define circular arcs. One such statement is used to place a fillet (or round) at the intersection of two lines. To illustrate, assume the following two line statements and the intersecting lines they produce:

```
LINE/ DX, 1
LINE/ DY, 1
```

To place a smooth arc at the intersection of these two lines, it is necessary to define only the desired radius:

```
LINE/ DX, 1
ARC/ .25
LINE/ DY, 1
```

There are many other ways to define geometric elements. A summary of the capabilities of the language is presented in the appendix of this manual.

A geometric statement may also describe the line class of an element. For example, in the statement:

```
DASHED, LINE/ DX, 5
```

the word DASHED indicates the desired line class for the line being defined. Figure 2 shows the line classes provided in the 1620 Drafting System and the language word for each.

Arithmetic Statements

Arithmetic statements aid the draftsman by performing computations needed to describe a drawing. For example, the following statement describes a circle with a center point named P1 and a diameter of 1.843 inches:

```
CIRCLE/ P1, 1.843/2
```

Because the radius, not the diameter, of a circle must be specified, it is necessary to divide 1.843 by two. This is indicated in the statement by the slash (/) and the digit 2. An arithmetic statement may also appear by itself:

```
A = B + C*D
```

This statement means: A is equal to B plus C times D.

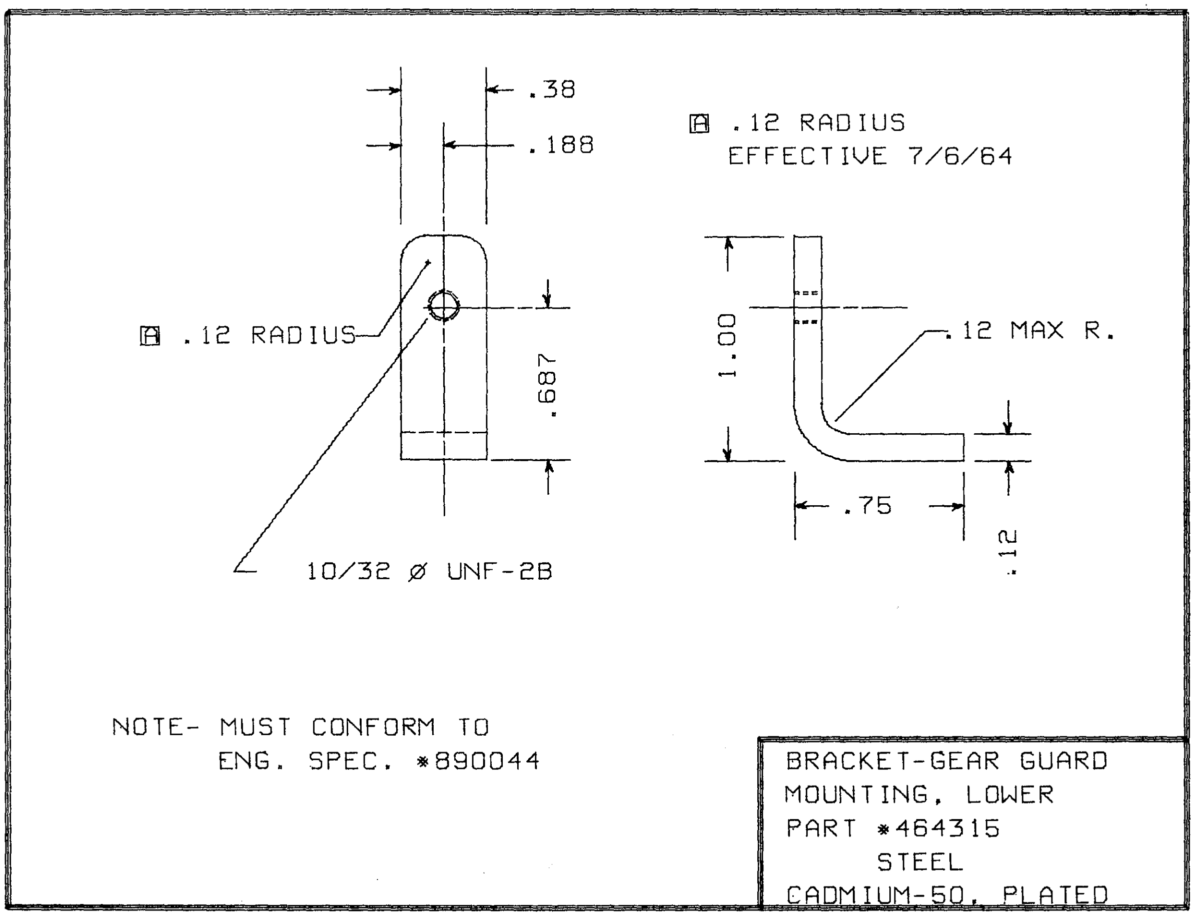

Figure 1.

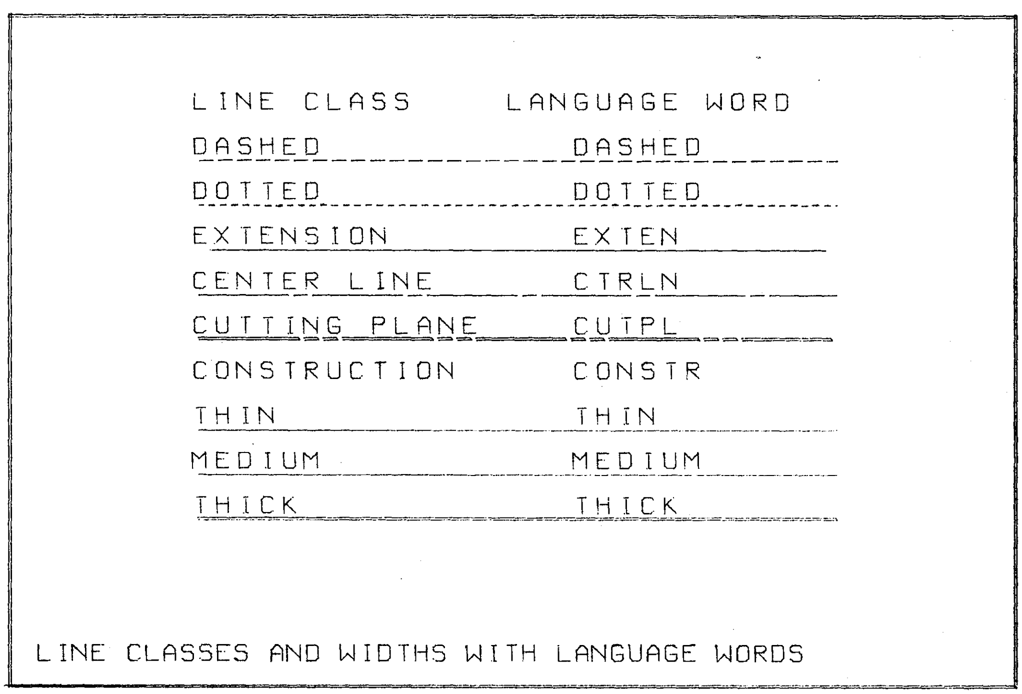

Figure 2.

Drawing Statements

Drawing statements produce notes, dimensions, and crosshatch lines.

A statement that draws crosshatch lines in a predefined area named PART is:

```
HATCH/ PART
```

Figure 3 illustrates the drawn output. The many lines inside the cross-sectional area are generated automatically by the 1620 Drafting System.

Dimension statements provide a variety of ways to dimension a drawing. The following statements produce the dimensioning shown in Figure 3:

1. MASK/ @D2@
2. DIMST/YSMALL, XCOMP, L1
3. DIM/ P1, P2
4. INDEX/ -1
5. DIM/ P2, P3
6. DIM/ P1, P4
7. DIM/ P1, P5, @7.00 REFERENCE@

Statement 1 determines the number of decimal places in the dimension value — in this case, two decimal places. The DIMST statement (2) starts a sequence of dimensioning; the statement indicates that X-component dimensioning is to start at the line labeled L1 and that successive dimensions are to be placed below L1. Statement 3 is the first command to dimension. This statement causes the following actions: The distance between the two points labeled P1 and P2 is calculated; extension lines are drawn down from P1 and P2; arrows are placed at the extension lines; and the calculated dimension is placed between the two arrows. Statement 4 places the next dimension on the same line as the previous dimension. Normally, the 1620 Drafting System spaces (or indexes) automatically between dimensions. The INDEX statement (4) orverrides the normal spacing. The dimensioning for statement 5, therefore, is placed along the dimension produced by statement 3. Statement 6 generates the dimension lines between the points P1 and P4. Statement 7 illustrates an option when drawing dimension lines. In this statement the desired text, instead of being generated by the drafting system, is placed within the dimension statement itself.

Figure 4 illustrates some ways of writing tolerances, choices of text alignment, and ways of displaying fractional, as well as decimal, dimensions.

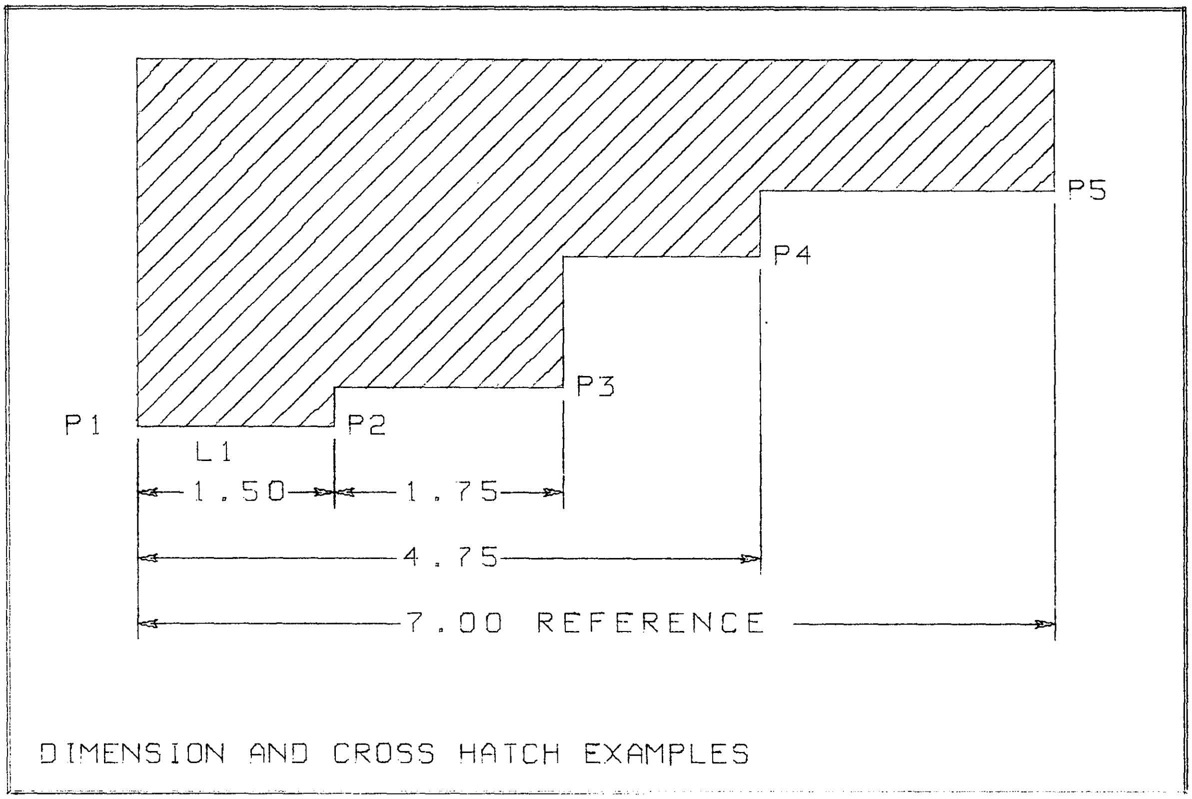

Figure 3.

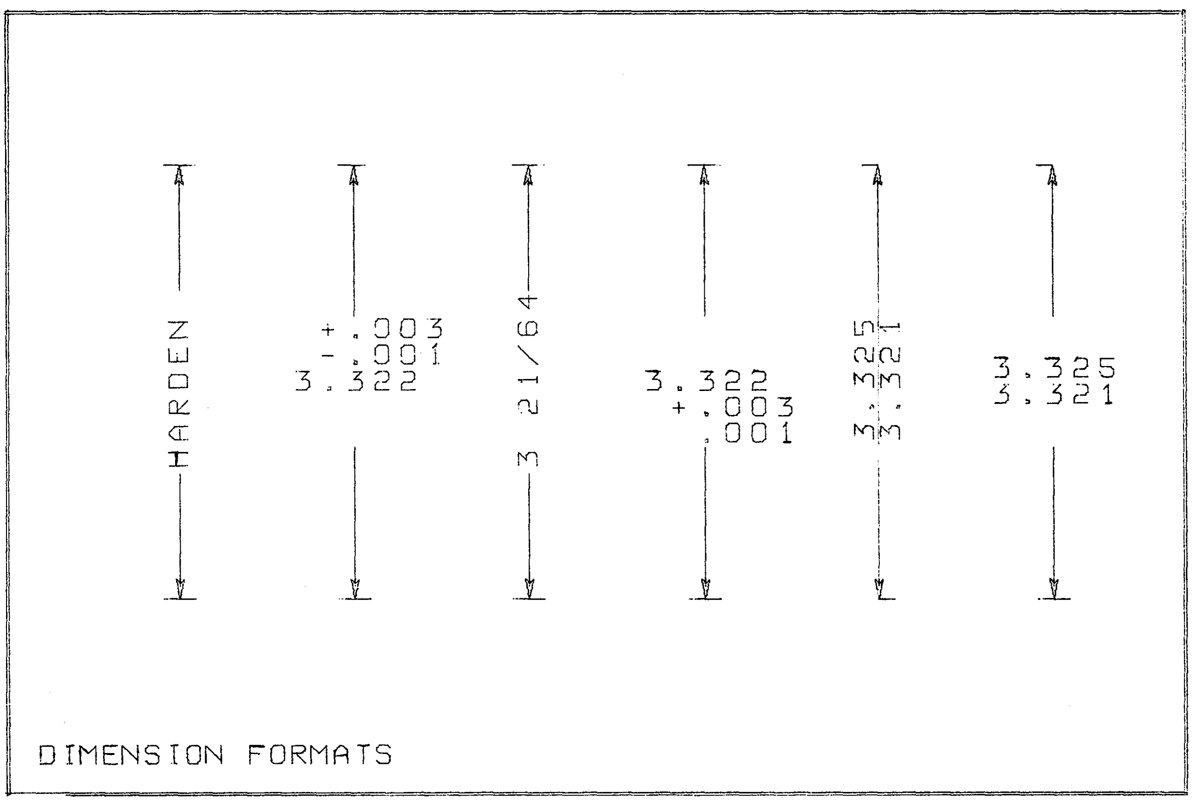

Figure 4.

Note statements allow the placement of local and general notes on drawings. The statement to cause the note and the arrow in Figure 5 is:

NOTE/ P1, 1, 1, @CHAMFER .05@, @2 PLACES@

The first value stated (P1) locates the arrow. The next two values (1, 1) indicate the length of the arrow and, in this manner, determine the location of the text.

CHAMFER .05
2 PLACES is the note to be written. Other note statements can be written to produce notes with arrows that begin to the right of the text. If no arrows are to be drawn, a third type of statement is written.

Control Statements

Control statements cause miscellaneous functions which aid the programmer in producing a drawing. One use is to define sections of a program in which looping occurs. The PAUSE statement is a control statement which can be used to print a message while a drawing is being processed. The following statement would cause the computer to stop during a drawing and to type out on the console the message CHANGE TO PEN 04.

PAUSE/ @CHANGE TO PEN 04@

During the pause the operator can change the pen and restart the drawing operation.

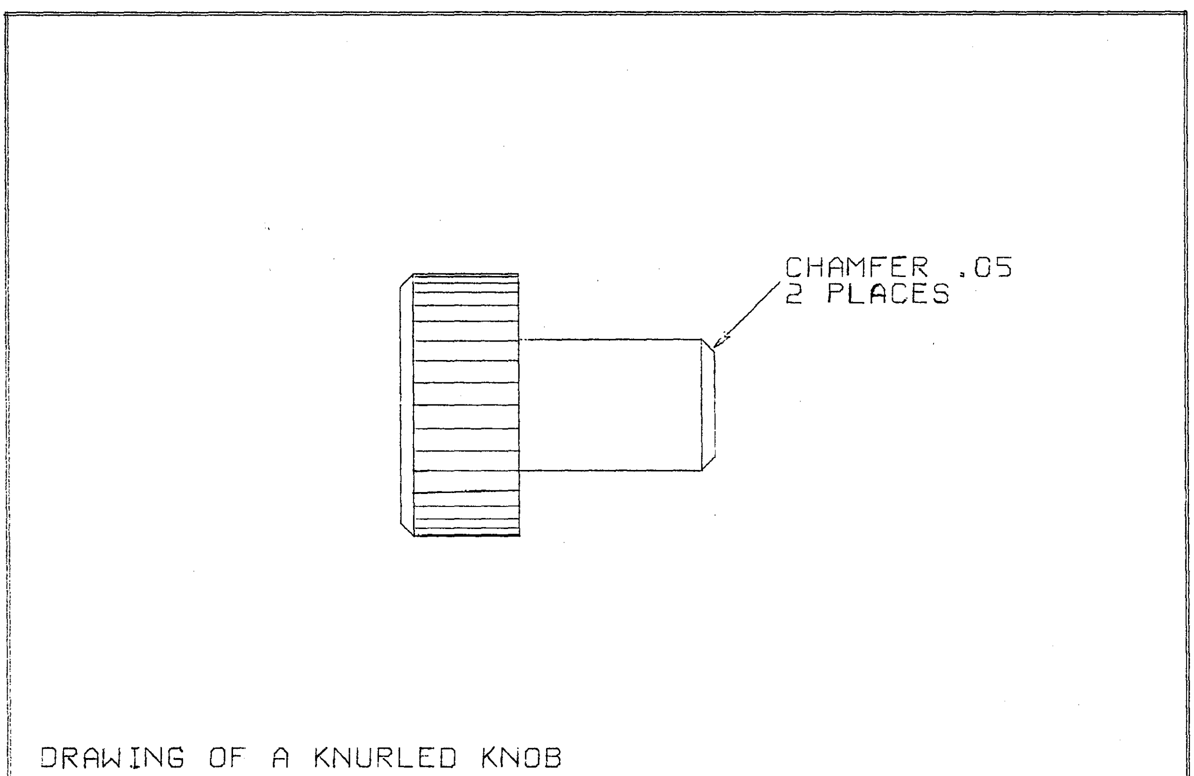

Figure 5.

GENERAL DESCRIPTION OF APPLICATION

Purpose and Objectives

The 1620 Drafting System is a practical method for the production of drawings. Features of the language make the system suited particularly for detailed engineering drawings; however, the general graphic capabilities of the language satisfy a wide range of drawing requirements. The advantages of the computer — high speed, precision, large storage capacity — are combined to reduce the effort required to produce drawings.

Application-Oriented Concepts

The language is designed specifically for drawing. Language statements enable many geometric relationships to be defined. Annotations, which can often be about one-half of the work on a drawing, are produced automatically by a variety of appropriate statements. Crosshatching and line-class generation are two other significant features of the language.

The language statements can be converted to a drawing quickly and simply. The processors for the 1620 Drafting System are stored on the disk pack and operate under the 1620 Monitor. An operator with a basic understanding of the procedures of the 1620 console need only add appropriate control cards to the language statement cards, ready the plotter, and begin processing. Drawing programs may be stacked together so that multiple drawings can be produced with no manual intervention.

Macros can be stored permanently in disk storage. Macros in the 1620 Drafting System are subparts of a drawing which occur frequently.

For example, a user of the 1620 Drafting System may draw spur gears frequently. All of the drawings of a part family such as this are essentially alike, differing only in dimensions. The general procedure for producing the gear drawing can be described in the drafting language, stored permanently on the disk pack, and recalled as needed by a single language statement that defines the appropriate dimensions and produces a unique drawing of the gear. The macro capability thus enables each user to customize the 1620 Drafting System to his own drawing requirements.

The 1620 Drafting System is designed so that it can become part of a larger, total engineering system. Any application in the engineering area which depends upon the coordinates of points — such as area or moment calculations — can be integrated by the user with the 1620 Drafting System. Geometric information about the drawn part can be stored on the disk pack and used by other application programs or can be outputted in the form of language statements. These language statements could, for example, be used as input to a numerical control program.

Extent of Coverage

The 1620 Drafting System is designed specifically to produce detail drawings. A detail drawing is one or more views of a single part. Such drawings are the normal output of drafting departments which support engineering design activities. However, the 1620 Drafting System can be used for other drawing requirements, some of which are:

1. Precision drawings. Drawings of this type take a great deal of time to produce manually because of the precision needed. The capabilities of a computer are suited ideally to precision drawings, such as comparator charts, master-tracer drawings, function displays, and master grids. (If the resolution of the IBM 1627 Plotter is not sufficient, the 1620 Drafting System can be modified readily to any other graphic output device.)

2. Standard or tabular drawings. Many companies produce families of drawings which differ from one another only by certain dimensions. Often it is necessary to produce a unique drawing for each new part within a family. The macro capability of the 1620 Drafting System makes it possible to produce such drawings in a fraction of the time needed by manual methods.

3. Graphic design drawings. Drawings of this type are the direct output of the design process. The arithmetic facilities of the 1620 Drafting System make it feasible to write design programs with direct graphic output. Drawings of this type include printed circuits, cams, tire treads, equipment layouts, container shapes, and pattern drawings.

4. Proposal drawings. In a proposal drawing, a company desires to tailor the use of common components to fit a specific customer's requirements. During the proposal cycle, drawings are highly susceptible to change. The 1620 Drafting System is suited particularly to all applications where changes in drawings are frequent and where common objects recur on drawings.

Advantages

Reduced drafting costs
Reduced drafting time
Ability to handle peak loads
Clear, accurate drawings
Information available for future processing

Machine-Oriented Concepts

The 1620 Drafting System uses the IBM 1311 Disk Storage drive with 1316 Disk Packs for programs, data, and user-created subdrawings. The direct access file greatly facilitates operation by permitting the 1620 Drafting System to convert language statements to a drawing in one monitored job.

The 1620 Drafting System has three processing phases: compiler, part processor, and drawing processor. The transition from one phase to another is handled automatically by the 1620 Monitor.

The compiler phase translates the language statements into 1620 instructions (object code) and writes the instructions onto the disk pack.

The part processor reads the object instructions back into the computer and executes the instructions. This phase solves all the geometric and arithmetic problems and writes out two files: a part model file and a drawing command file.

The drawing processor, after reading a drawing command record, accesses the part model file and converts it to a drawing on the 1627 Plotter. This phase also develops the notes, dimension lines, and cross-hatching lines that appear on the drawing.

Timing

The total time for the computer to produce an average A-size drawing (8.5" x 11") is about 10 to 15 minutes.

The factors that determine running time are (1) number of input statements, (2) total length of line work on the drawing, (3) length of moves in going from one line to another, and (4) number of alphabetic characters to be drawn. Generally, the actual drawing operation on the plotter represents the greater part of the overall process time.

The plotters draw at a rate of two to three inches per second. Alphabetic characters are drawn at one character per second. The average compiling speed on a 1620 Model 1 is one card per second.

Special Techniques

A model-oriented syntactical compiler is used to convert the input language statements to an object program; the compiler is both efficient and fast. The language of the 1620 Drafting System is contained within tables in the compiler, so that the user may modify the system easily to amend or to extend the capabilities of the language.

A two-dimensional part model file, consisting of bounded geometric elements, is developed and stored on the disk. This file can be used in other applications — for example, area, center of gravity, or moment of inertia calculations.

In all phases, the processing programs of the 1620 Drafting System are loaded into storage in core-image format to permit the fastest possible operation.

The storage of user-created macros on the disk pack is a feature of the 1620 Drafting System. These macros become a permanent part of the user's system, and they are stored as actual object instructions.

Restrictions and Range

The part model file has capacity for about 1500 object lines. The lines for dimensioning and notes are not resident on the disk file and are not a part of the object line limit.

About 100 macros can be stored on the disk file.

Precision

Computations involving part coordinates are performed in eight-place, floating-point arithmetic. Drawing (or paper) coordinates are manipulated in fixed-point arithmetic, with four decimal places.

MACHINE AND SYSTEMS CONFIGURATION

Machine Configuration

IBM 1620 Data Processing System, 20K, Model 1
Additional Instructions (TNS, TNF, MF)
Automatic Divide
Indirect Addressing
1622 Card Read Punch and Card Read Punch Adapter
1311 Disk Storage Drive, Model 3, and Disk Storage Drive Adapter
Paper Tape Adapter
1627 Plotter with 1626 Plotter Control

Or:

IBM 1620 Data Processing System, 20K, Model 2
1622 Card Read Punch and Card Read Punch Adapter
Paper Tape Adapter
1627 Plotter with 1626 Plotter Control.

Programming Systems

IBM 1620 Monitor I (1620-PR-025), Version I, modification level 9. Monitor dim numbers 0810 through 0879 are used by the 1620 Drafting System.

GENERAL SYSTEMS CHART OF APPLICATION

The procedure for producing a drawing with the 1620 Drafting System is illustrated in Figure 6. The engineering sketch or layout drawing is broken down into views which adequately describe the part. The engineer or draftsman describes each view with statements from the language of the 1620 Drafting System. Some of the statements describe graphic elements (such as lines and arcs), while others describe groups of elements which have been stored previously on the disk file (such as cap screws and bolts). The writing of the statements parallels the actions of a draftsman drawing the part manually.

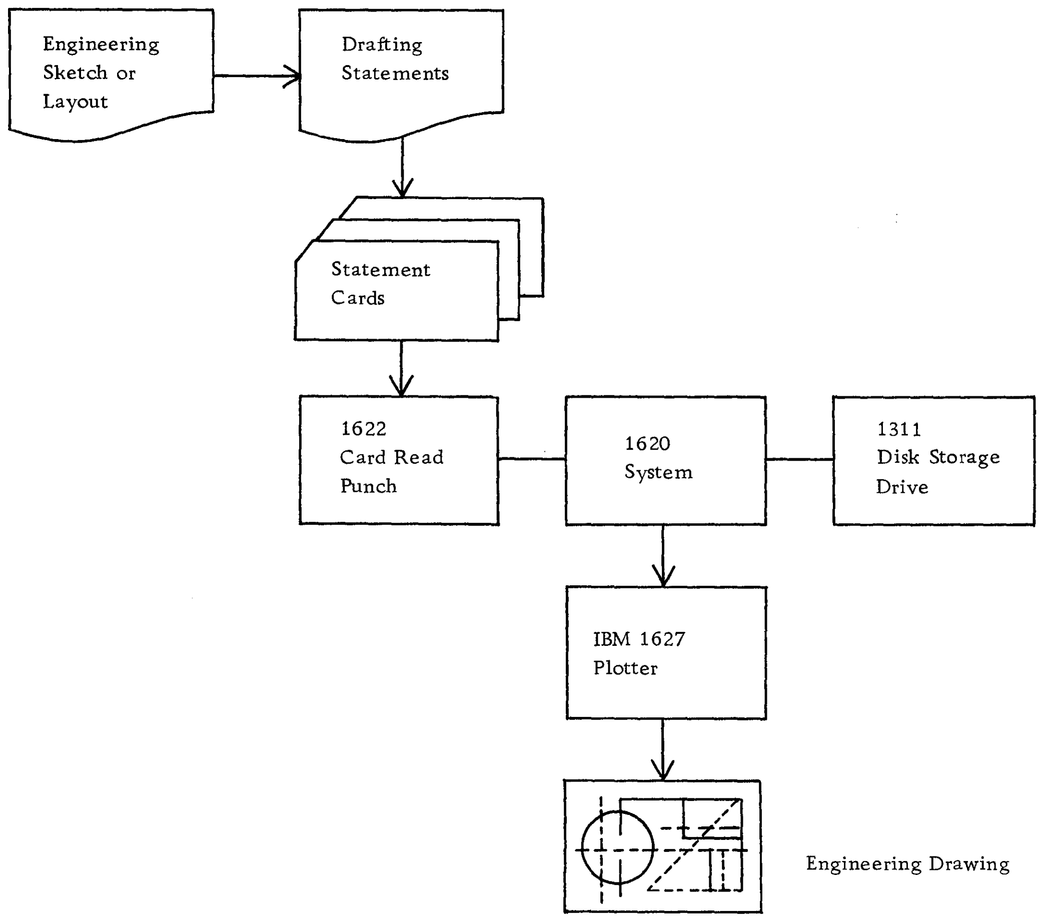

Figure 6.

The statements are punched into cards and entered into the computer. The disk-stored processor programs are read into the 1620 and used to translate the statements and to control the IBM 1627 Plotter to produce a drawing. The drawing is then sent back to the coder for final review and approval before it is released to other departments. Changes can be made in the drawing by correcting appropriate statements and by re-processing all statements to produce a new drawing.

LIST OF INPUT/OUTPUT FILES

File Name	Reference
Drafting Language Statements	Programmer's Manual
Part Model File	System Manual
Drawing Command File	System Manual
Macro File	System Manual

SAMPLE PROBLEM

Figure 7 is an engineering drawing of a switchgear frame. Figure 8 illustrates the coding necessary to produce the drawing. The comments after the two dollar signs are not required, but are included to explain the coding. Statements 0007 to 0019 illustrate geometric definitions of object lines. The arithmetic features of the language are shown by statements 0024 to 0028. Statements 0045 to 0066 produce the dimensioning on the drawing. The notes on the drawing are caused by statements 0067 to 0072.

Coding of the drawing took 40 minutes. The total time for the computer to process the drawing on a 1620 Model I with a 1627 Plotter Model 1, was 7.8 minutes.

The operating instructions to run the sample problem are contained in the Application Directory.

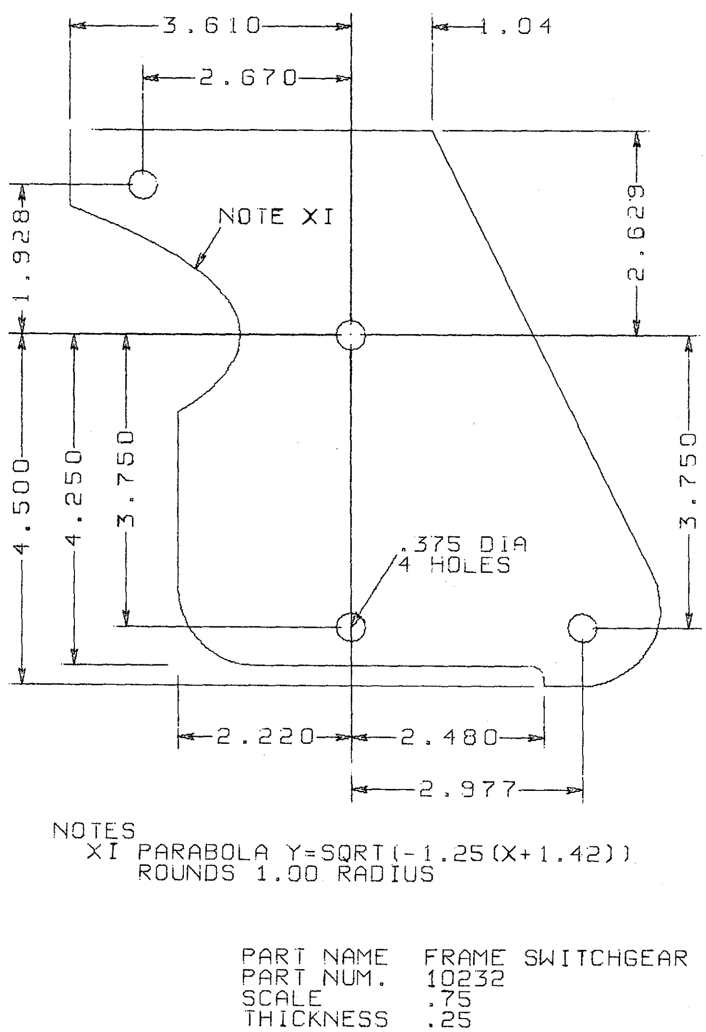

Figure 7.

```
*LIST PRINTER
$$ SAMPLE PROBLEM FRAME-SWITCHGEAR   1620 DRAFTING SYSTEM
$$
      ORG=LINE/0,0,0,0 $$ REF LINE USED TO AID DIMENSIONING
        C5=CONSTR,CIRCLE/2.977,-3.5,1      $$ CONSTRUCTION CIRCLE
   FRAME=VIEW/                        $$ START PART DESCRIPTION
      HOLE=.375/2                     $$ HOLE RADIUS
        C1=CIRCLE/0,-3.75,HOLE        $$ FOUR
        C2=CIRCLE/2.977,-3.75,HOLE    $$     HOLES
        C3=CIRCLE/0,0,HOLE            $$       THE SAME
        C4=CIRCLE/-2.67,1.928,HOLE    $$           SIZE
        L1=LINE/-3.61,2.629,1.04,2.629     $$  TOP LINE
        L2=LINE/PPP,LEFT,TANTO,C5          $$  PPP IS LAST END POINT
            ARC/PPP,2.977,-4.5,XSMALL,RADIUS,1.0,CLW
        L3=LINE/2.977,-4.5,2.48,-4.5
        L4=LINE/PPP,2.48,-4.25
            ARC/.25/2                 $$ SMOOTH AN ARC BETWEEN TWO LINES
        L5=LINE/2.48,-4.25,-2.22,-4.25
            ARC/1.0                   $$ SMOOTH AN ARC BETWEEN TWO LINES
        L6=LINE/-2.22,-4.25,-2.22,-SQRT(-1.25*(-2.22+1.42))
$$
$$           DEVELOP CURVE OF THE PARABOLA WITH SHORT LINE SEGMENTS
$$
           LOOPST/                    $$ START LOOP
      YHI=+SQRT(-1.25*(-3.61+1.42))   $$ MAX Y VALUE OF THE PARABOLA
      YLO=-SQRT(-1.25*(-2.22+1.42))   $$ MIN Y VALUE OF THE PARABOLA
      YINC=YLO                        $$ SET INCREMENT MIN Y VALUE
1)    YINC=YINC+.1                    $$ INCREMENT Y VALUES OF PARABOLA
      XINC=-(YINC*YINC+1.25*1.42)/1.25  $$ COMPUTE X FOR GIVEN Y
           IF(YHI-YINC)3,2,2            $$ TEST IF PARABOLA COMPLETE
                                        $$   GO TO 2) IF NOT COMPLETE
                                        $$   GO TO 3) IF COMPLETE
2)         LINE/PPP,XINC,YINC           $$ LINE SEGMENTS OF PARABOLA
           JUMPTO/1                     $$ GO TO 1) NEXT LINE SEGMENT
3)         LINE/PPP,-3.61,YHI           $$ LAST PARABOLA LINE SEGMENT
           LOOPND/                      $$ END LOOP
$$
        L7=LINE/PPP,-3.61,2.629
           END/FRAME                    $$ END OF THE VIEW
$$
$$ END PART DESCRIPTION,START DRAWING,DIMENSIONS AND NOTES
$$
           SCALE/ .75                   $$ SCALE THE PART
           ORIGIN/ 3.5, 7               $$ REFERENCE THE PART TO THE PAPE
           DRAW/ FRAME                  $$ DRAW THE PART
         MASK/'D3'                         $$ DIM TO THREE DECIMALS
         DIMST/YSMALL,XCOMP,L3             $$ DIMENSION AT THE
         DIM  /ORG,L6                      $$     BOTTOM OF THE PART
         INDEX/-1                          $$ DIM ON SAME LEVEL AS LAST
         DIM  /ORG,L4
         DIM  /ORG,(POINT/C2)
         DIMST/XLARGE,YCOMP,(LINE/3,0,3,4) $$ DIMENSION AT THE
         DIM  /ORG,L1                      $$   RIGHT OF THE PART
         DIM  /ORG,(POINT/C2)


         DIMST/XSMALL,YCOMP,L6             $$ DIMENSION AT THE
         DIM  /ORG,(POINT/C1)              $$     LEFT OF THE PART
         DIM  /ORG,L5
         DIM  /ORG,L3
         INDEX/-1                          $$ DIM SAME LEVEL AS LAST
         DIM  /ORG,(POINT/C4)
         DIMST/YLARGE,XCOMP,L1             $$ DIMENSION AT THE
         DIM  /ORG,(POINT/C4)              $$    TOP OF THE PART
         DIM  /ORG,L7
         INDEX/-1                          $$ DIM SAME LEVEL AS LAST
         DIMP /2,.2,2,2,                   $$ REVERSE DIM ARROWS
         MASK/'D2'                         $$ DIM TO TWO DECIMALS
         DIM  /ORG,(POINT/L1)
NOTE/-2,SQRT(-1.25*(-.58)),.3,.5,'NOTE XI@
NOTE/0,-3.75,.5,.8,'.375 DIA','4 HOLES'
TITLE/.7,2.2,'NOTES',' XI PARABOLA Y=SQRT(-1.25(X+1.42))@,$
'    ROUNDS 1.00 RADIUS'
TITLE/2.5,1,'PART NAME  FRAME SWITCHGEAR @,@PART NUM.  10232 @,$
'SCALE      .75 ','THICKNESS  .25'
         FINI/
```

Figure 8.

APPENDIX -- LANGUAGE SUMMARY

This summary of the language of the 1620 Drafting System covers the major features of the system; not every statement is included. In this discussion all language words of the 1620 Drafting System are in uppercase letters; where multiple words appear stacked in a statement — such as LEFT / RIGHT — one word only is to be selected when a statement is written.

The underlined, lowercase words represent variable information; the text to the right of each word or group of words defines the use in the language of the 1620 Drafting System:

scalar	A number or label of a number, or an arithmetic expression that results in a number or the label of such an expression.
point, line, circle, arc	A label of the indicated geometric element or a phrase that results in such an element.
point2 line2 circle2	A label of the indicated geometric element or a phrase, within parentheses, that results in such an element.
view	A label of a VIEW definition.
shape	A label of a SHAPE definition.

Point Definitions

Rectangular coordinates	POINT/ scalar, scalar
Intersection of two lines	POINT/ INTOF, line, line2
Second point of a line	POINT/ line
Center point of a circle	POINT/ circle
	POINT/ CENTER circle
Center point of an arc	POINT/ arc
	POINT/ CENTER, arc

Line Definitions

From a point to a point	LINE/ point, point
Relative distance	LINE/ point, DX / DY scalar
Distance from present part position	LINE/ DX / DY, scalar
From a point at an angle to the X-axis	LINE/point, ATANGL, scalar, TILLX / LENGTH / TILLY, scalar

Intersection of a line and a circle

```
          XLARGE
LINE/XSMALL, INTOF, line, circle2
          YLARGE
          YSMALL
```

Intersection of two circles

```
          XLARGE
LINE/XSMALL, INTOF, circle, circle2
          YLARGE
          YSMALL
```

From a point tangent to a circle

```
                  LEFT
LINE/ point, RIGHT, TANTO, circle
```

Tangent to two circles

```
      LEFT                      LEFT
LINE/RIGHT, TANTO, circle, RIGHT, TANTO, circle
```

Parallel to a line through a point

```
LINE/ point, PARLEL, line
```

Parallel to a line offset a distance

```
          XLARGE
LINE/XSMALL, PARLEL, line, scalar
          YLARGE
          YSMALL
```

Perpendicular to a line from a point

```
LINE/point, PERPTO, line
```

Circle Definitions

Center point and radius

```
CIRCLE/ point, scalar
```

Defined by three points

```
CIRCLE/ point, point, point2
```

Tangent to two lines

```
            XLARGE               XLARGE
CIRCLE/XSMALL, TANTO, line, XSMALL, TANTO, line, scalar
            YLARGE               YLARGE
            YSMALL               YSMALL
```

Center and radius of an arc

```
CIRCLE/ arc
```

Arc Definitions

Center point, radius, starting angle, and sweep angle

```
ARC/point, scalar, scalar, scalar
```

Between two points

```
                   XLARGE
ARC/ point, point, XSMALL, RADIUS, scalar, CLW
                   YLARGE
                   YSMALL
```

Tangent to two lines (Filleting)

```
ARC/scalar
```

Geometric Functions

Delta X value of a line	DXOF (line)
Delta Y value of a line	DYOF (line)
Length of a line	DIST (line)
Mirror of a point	MIRX (point)
	MIRY (point)
	MIRXY (point)
Angle of a line	ATAND (line)
Extract a parameter, where n is the parameter number of the standard form. The radius of a circle could be: PARAM (3, C4)	PARAM (n, point line circle arc)

Line Class

A line class can be assigned to a single geometric definition or to a group of definitions. The following line classes are a part of the system:

Line Class	Language Word
Construction line	CONSTR
Center line	CTRLN
Cutting-plane line	CUTPL
Dashed line	DASHED
Dotted line	DOTTED
Extension line	EXTEN
Thin line	THIN
Medium line	MEDIUM
Thick line	THICK

Auxiliary Part Reference System

Translation	REFSYS/ point
Translation and rotation	REFSYS/ point, ATANGL, scalar

Geometric Grouping

Group geometric elements into a view.	label=line class, VIEW/
Group geometric elements into a shape which is part of a view.	label=line class, SHAPE/
End view or shape definition.	END/ view shape

Arithmetic Statements

The following arithmetic operations can be performed by the 1620 Drafting System:

Operation	Coding
Addition	+
Subtraction	-
Multiplication	*
Division	/
Exponentiation	**
Sine of angle in degrees	SIND (scalar)
Cosine of angle in degrees	COSD (scalar)
Arctangent in degrees	ATAND (scalar)
Absolute value	ABS (scalar)
Square root	SQRT (scalar)
Natural log	ALOG (scalar)
Exponential	EXP (scalar)

Sample arithmetic statement: YHI=-SQRT (-1.25* (-2.22 + 1.42))

Drawing Reference System

Scale the part	SCALE/ scalar
Scale the part by an X and a Y scale factor	SCALE/ scalar, scalar
Translate to paper coordinates	ORIGIN/ point
Rotate drawing	ORIGIN/ point, ATANGL, scalar

Drawing Action Statements

Draw the part	DRAW/ shape view dim number
Draw a mirror image	DRAW/ MIRX MIRY (view or shape) MIRXY
Crosshatch a drawing	HATCH/ view shape

Note Statements

Place an arrow from the part point to the text. The scalars are delta X and Y paper distances.	NOTE/point, scalar, scalar, @ANY TEXT@
A note with the arrow from the right side of the text.	NOTER/point, scalar, scalar, @ANY TEXT@
A note with no arrow.	NOTE/ point, @ANY TEXT@
A general note at a paper location.	TITLE/ point, @ANY TEXT@

Dimension Statements

Start dimensioning from the given line in either component or parallel dimensions. Automatically index in the direction shown by one of the first four words. The optional scalar can be used to set the indexing distance.	DIMST/XLARGE, XSMALL, YLARGE, YSMALL, XCOMP, YCOMP, TRUE, line, scalar
The dimension commands which produce extension lines, arrows, and the calculated distance. The optional text is used in place of a calculated distance. Extension line control is obtained by the alternate words: DIMEN, DIMNE, DIMNN, DIMEE, where the terminals E and N produce either an Extension or No extension line. (DIM and DIMEE are equivalent.)	DIM/ point, point, @OPTIONAL TEXT@ DIM/ line, point2, @OPTIONAL TEXT@ DIM/ line, line2, @OPTIONAL TEXT@
Alter the automatic indexing; INDEX/ 1 causes a space to be skipped.	INDEX/ scalar
The MASK statement determines the format of the dimensioning text. The two scalars can be used for tolerance values.	MASK/ scalar, scalar, @FORMAT CODES@

Systems Control Statements

Type a message and pause.	PAUSE/@ANY MESSAGE@
Define an abbreviation for a statement, phrase, or word. For example: XL=DEFINE/@XLARGE, @ allows the programmer to use XL instead of XLARGE.	label=DEFINE/ @ANY PHRASE@

Looping Control Statements

Begin a loop.	LOOPST/
End of a loop.	LOOPND/
Go to statement.	JUMPTO/ statement number
Arithmetic IF statement	IF (scalar) 3 statement numbers

Macro Control Statements

Begin a macro definition	label=MACRO/ parameter names
Terminate a macro definition	TERMAC/
Call a macro for execution. For example, to call a macro named RECT with parameters labeled LEN and WID, write: CALL/RECT, , LEN=2, , WID=3.4	CALL/macro-name, , parameter definitions, , . . .

H20-0078-1

Printed in U.S.A. H20-0078-1

IBM
International Business Machines Corporation
Data Processing Division
112 East Post Road, White Plains, New York 10601