Business Graphics Applications Using DSG/3000

Cecile Chi

The value of graphs as an effective method of presenting information is widely acknowledged. As a result, many graphs are being produced manually or at service bureau charges are growing rapidly. The HP3000 with the Decision Support Graphics (DSG) software package to the rescue!

Some graphics requirements are of the one-time, type-in-the-data variety, using either paper or transparencies. These are drawn by managers, professionals, or secretaries using the Multiplot package on the HP2647/2648 standalone graphics system or DSG/3000. My experience has been that people are not willing to take the time to read through the booklets and try the examples in the Self-Paces DSG course. It's always "Just show me how to start this thing!" followed by "What does it want now?" and "How do I tell it to do what I want?" Half an hour of explanation and question-answering is usually enough to get a new user going on the first application, and after that it's a matter of being available to answer questions from time to time. A major advantage of DSG over Multiplot is that neither graph design nor data disappear when the user goes on to the next graph or turns off the terminal. A one-time graph is seldom really a one-time graph. Either another copy is required, or an updated version with new data is requested, or the chart definition is used as the basis for a revised design or for the next graph to be designed. People are more productive when the graph and data are ready and waiting to be redrawn or changed.

The occasional or periodic user sometimes requests documentation of the chart design already developed and saved in order to plan the next revision while waiting for the terminal to become available, or to refer to while designing another chart on the screen. A 2631G printer attached to the 2647A graphics terminal via the HP-IB is used for this purpose. The DSG screens can be transferred to the printer by using the 2647A's Command mode. Before enabling Command mode, the cursor must be positioned in the upper left corner of the screen. Homing the cursor will get it to the first unprotected field, and using the cursor arrow keys to get it to the beginning of the window will make it possible to print the screen title. Pressing the CNTL key and the 55 key simultaneously releases the protected fields, making it possible to copy them to the printer. The sequence required to print a copy of the screen is

```
COMMAND
COPY(f2)
ALL(f3) from
DISPLAY(f3)
to HP-IB(f7)#
6 (or whatever your printer address is)
RETURN.
```

Pressing the COMMAND key disables Command mode, and pressing the CNTL key and *f*4 key simultaneously turns on the V/3000 field protection, getting you back into DSG. It is necessary to get out of Command mode and into the DSG in order to get to the next screen to be printed. Paging of the printer must be done manually, since the Copy command does not cause a page eject. Two screens will fit nicely on a page, with a few blank lines to separate them. Similary, the graph drawn on the screen can be transferred to the printer using the sequence

```
COMMAND
next(f1)
next(f1)
TRANSFER(f3)
ALL(f3) from
GRAPHICS(f4)
to HP-IB(f7)#
6(printer address)
RETURN;
```

Use COMMAND to get back to DSG. The Transfer command does cause a page eject on the printer at the end of the graph. We end up with documentation which fits on 8½"×11" paper, ready to go into the user's file folder or loose-leaf notebook.

DSG contains an option (which is implemented from the Main Control menu) to draw each graph in a chart file, with or without a pause between graphs. The graphs may be directed to either the screen of the graphics terminal or to the plotter. Drawing all of the charts from a file to the plotter requires either an operator to change the paper between plots or a scrolling option on the plotter to advance the roll of plot paper between graphs.

Another option for unattended production of the series of graphs contained in a chart file is to use the printer. Printer graphs are black and white, without anywhere near the resolution of a plotter graph, but for screening or quick reference they are frequently worth the savings in time and cost. The GRAFPRNT program on the swap tape for this conference is a generalized graph-printing program. It is run from a graphics terminal which has a graphics printer (2631G) attached via the HP-IB; it temporarily resets the device destination and chart size and then produces all charts in any given chart file on the terminal screen and then transfers them to the printer. The COBOL source code is included, since the compiled version includes directions to a 2647A graphics terminal and a printer on HP-IB address 6; these may need to be changed to fit your hardware. A pause is required to allow time for transferring the graph from the screen to the printer; the call to "FPAUSE" calls a FORTRAN subroutine which calls the PAUSE intrinsic. The PAUSE intrinsic can't be called from COBOL because it requires a REAL parameter, and HP COBOL does not support data type REAL. The 40-second pause used in this program has been adequate for all applications I've implemented so far; it may not be the optimum length of time.

The methods discussed so far have required designing each individual graph using DSG. However, there are many applications in the business world which require sets of graphs identical in design except for a few variables. One variable is usually a title line, and another would probably be either the data file name or the data subset specification. Labels and additional title lines may also be variables. An example of this method is the Sales Graph system illustrated with a flowchart (Exhibit 1) and UDC:

```
SALEGRAF
FILE TITLES=SALENAME
FILE DATAFILE=SALEDATA
RUN EXTRACT;PARM=%100000
FILE GRAFFILE=DEMOGRAF
RUN DRAW;LIB=G;PARM=%100000
***************************
ARGRAF
FILE TITLES=ARNAME
FILE DATAFILE=ARDATA
RUN EXTRACT;PARM=%20000
FILE GRAFFILE=DEMOGRAF
RUN DRAW;LIB=G;PARM=%20000
***************************
```

The title file, which may be created and maintained with EDIT/3000, contains the graph titles. It also specifies the plants for which data is to be extracted from the database and controls the order in which graphs are drawn. The EXTRACT program retrieves the specified data for the plants listed in the title file and arranges it in appropriate format for DSG. The PARM parameter on the RUN command sets a software switch which is used by the EXTRACT program to identify the account code to be used for extracting data from the database, and by the draw program to identify the chart name to retrieve from the chart file. This program may be replaced by QUERY if the database is designed in such a way that QUERY is adequate, or by AQ if it is implemented at your site and will handle your needs. The major limitation of QUERY is that it can access only one data set at a time. AQ can access and concatenate multiple data sets, and the added flexibility may make it possible to use it in place of an Extract program. Of course, if you need a linear regression on your data, you need a program. Using whichever method is most appropriate, a date file is created, containing data in DSG format.

Then a graph drawing program, DRAW, is initiated by the UDC. This program retrieves the appropriate chart from the chart file and then reads through the title file. There are optional methods of writing this program, depending upon whether the plotter being used has the scrolling option. If it is a scrollable plotter, the program can just cycle through the title file, setting the title and data subset specifications and drawing a graph for each record in the title file. If the scrolling option is not available on the plotter being used, the program is written to send a message to the terminal each time it reads a record in the title file, asking the operator whether it should draw a graph for the current title and reminding him or her to change the paper on the plotter.

Some applications consist of groups of very similar charts, such as graphs of budget versus actual over time for each of many departments or divisions, for each of a number of measures. Multiple copies are requried in many cases. Since the quantities run to dozens or hundreds per monthly batch, it is not practical to draw them on a pen plotter without a scrolling option, which requires someone to change the paper after each plot. Color plots and high resolution are not necessary for screening large numbers of graphs, looking for trends and exceptions, so the cost of a scrollable plotter would not be considered justifiable.

The next option handles the batches of graphs which need to be repeated for multilpe groups and multiple measures. The method illustrated in Exhibit 2 uses data which is retrieved through RJE from a non-HP mainframe using a report tabulating utility program. The utility is run on the mainframe to produce a "printed" report according to pre-defined specifications such as no titles and the use of identification codes as column headers, with the report routed to the HP3000. Then the report is read onto a disc file using RJE. A COBOL program is used to read this disc file and flip the matrix, producing a data file in a format suitable for DSG and printing a format listing of the data file to be used for data definition. At this point, the user designs a set of graphs in one chart file, using DSG, with the graphs in the same order as the data. The title file, containing the identification codes to be used for data subset specifications and the variable title line, is arranged in the order in which the graphs should print. Then the graph-drawing program MULTGRAF is initiated. This program cycles through the chart file, drawing a graph for each location for each chart. If multiple copies are requested, it starts over. More efficient utilization of the computer would be achieved by doing multiple transfers of each screen; in our case, it was decided to use more machine time to redraw the additional copies in order to avoid the clerical time required to sort the graphs. By printing them in the proper order, the whole run can be fed through a burster and then separated into groups for attachment to reports. Once the data file has been created and the chart designs done, any given individual graph can be selected interactively by setting the subset specification and the title on the DSG menus and then drawing the graph on the screen and transferring to the printer or drawing it on the plotter to get colors and better resolution.

Our largest monthly production run so far consists of 4 copies of graphs for 17 locations on 18 to 20 different measures, for a total of over 1200 graphs. This run takes over 24 hours, so it is either run on a weekend or run with 2 copies on two nights. Even a scrollable plotter would require approximately 5 minutes per graph, or 100 hours, which is not acceptable turn-around time. The best service bureau bid, based on large volume regularly, was $10 per graph and 4-day turnaround.

A frequent question from management or professional users is, "Can't you print the data under the graphs?"; the answer is "Well, yes, but it requires some custom programming." The method illustrated in Exhibit 3 requires two programs and two passes of the paper through the printers. The REPORT program generates reports with about eight lines of title and heading information, a lot of blank space, and then the labeled lines of data, and footnotes. This program also creates the title file and data file for DSG. If the application did not require so many heading lines, the titles could be printed from DSG rather that the Report program. The paper containing the reports is fed into the graphics printer, and another version of MULTGRAF is used to print the graphs in the blank spaces on the reports. It would be possible to draw plotter graphs on the reports instead of printing graphs, by setting the graph dimensions appropriately and changing the paper after each graph. Keeping everything in the right order would require an operator's undivided attention, however, and thus is probable not practical for production runs of any size.

The methods of utilizing DSG programatically described here are early experiments at improving office productivity by providing better tools. The fact that these tools are being used as fast as they are developed demonstrates that they fill a need, and should encourage continued development of better tools. The release by the HP lab of their Contributed Graphics Library (CGL/3000) or graphics intrinsics will provide programmers with the opportunity to develop new sets of tools.

EXHIBIT 1

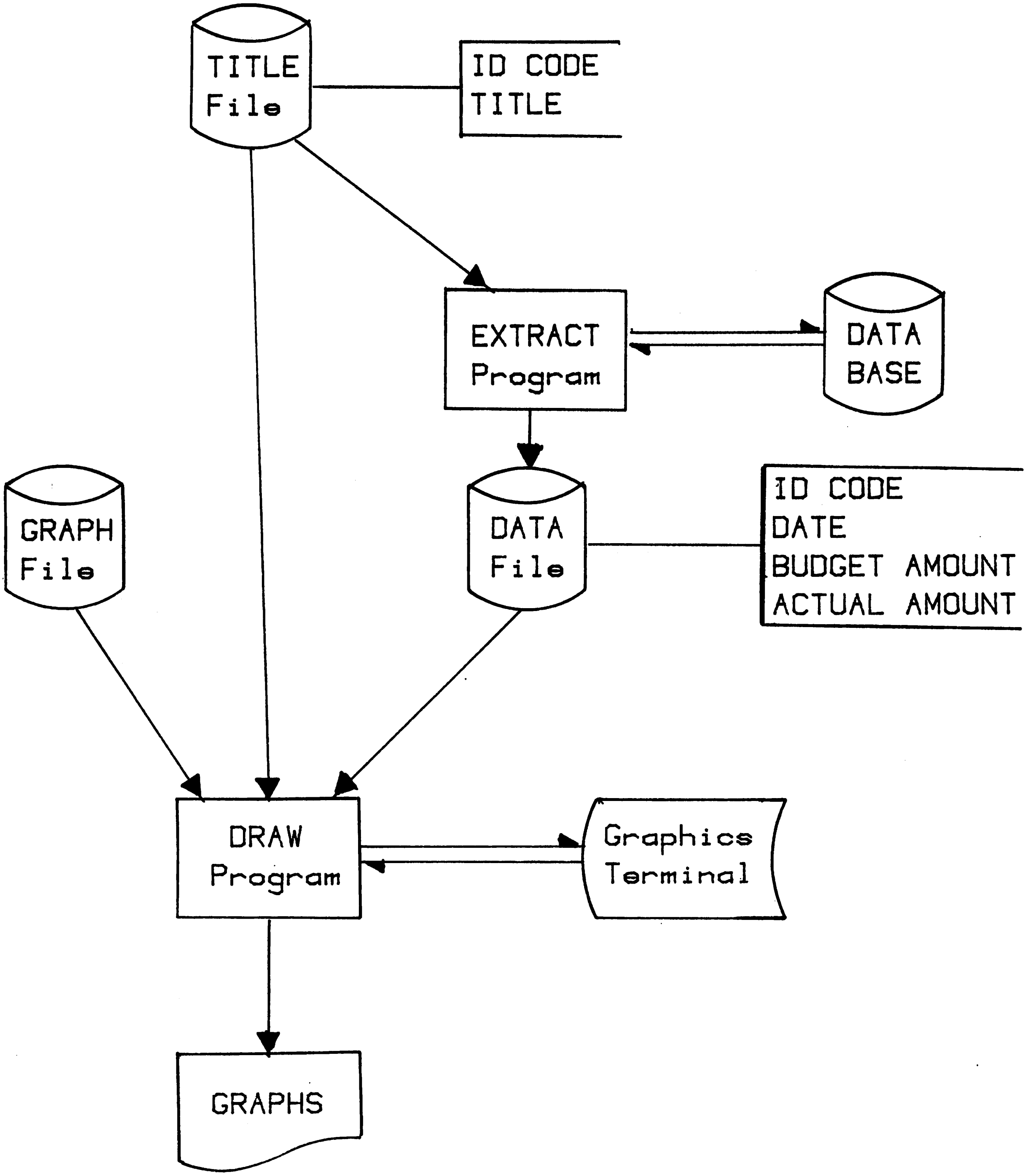

TITLE File
ID CODE
TITLE
EXTRACT Program
DATA BASE
GRAPH File
DATA File
ID CODE
DATE
BUDGET AMOUNT
ACTUAL AMOUNT
DRAW Program
Graphics Terminal
GRAPHS

EXHIBIT 2

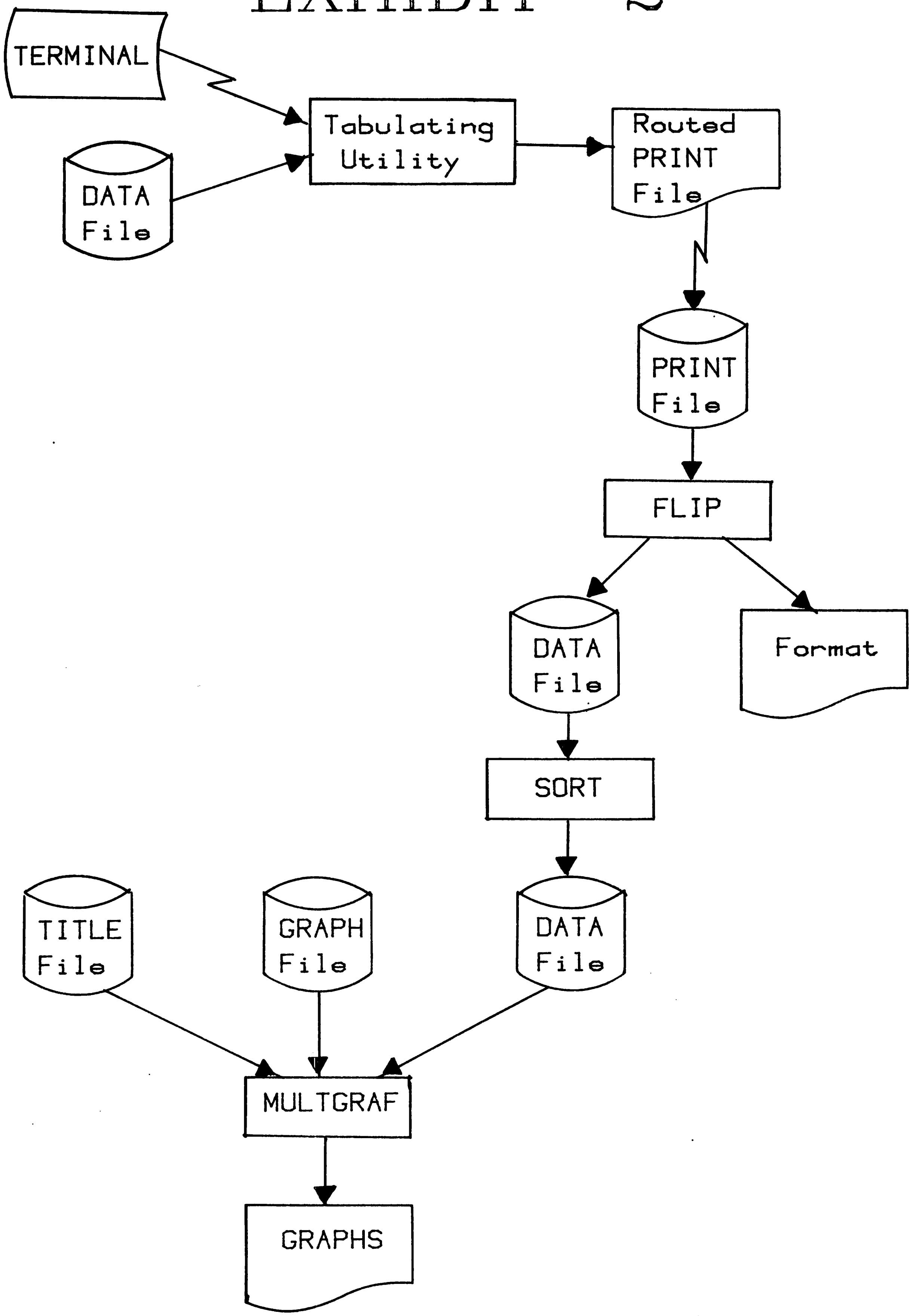
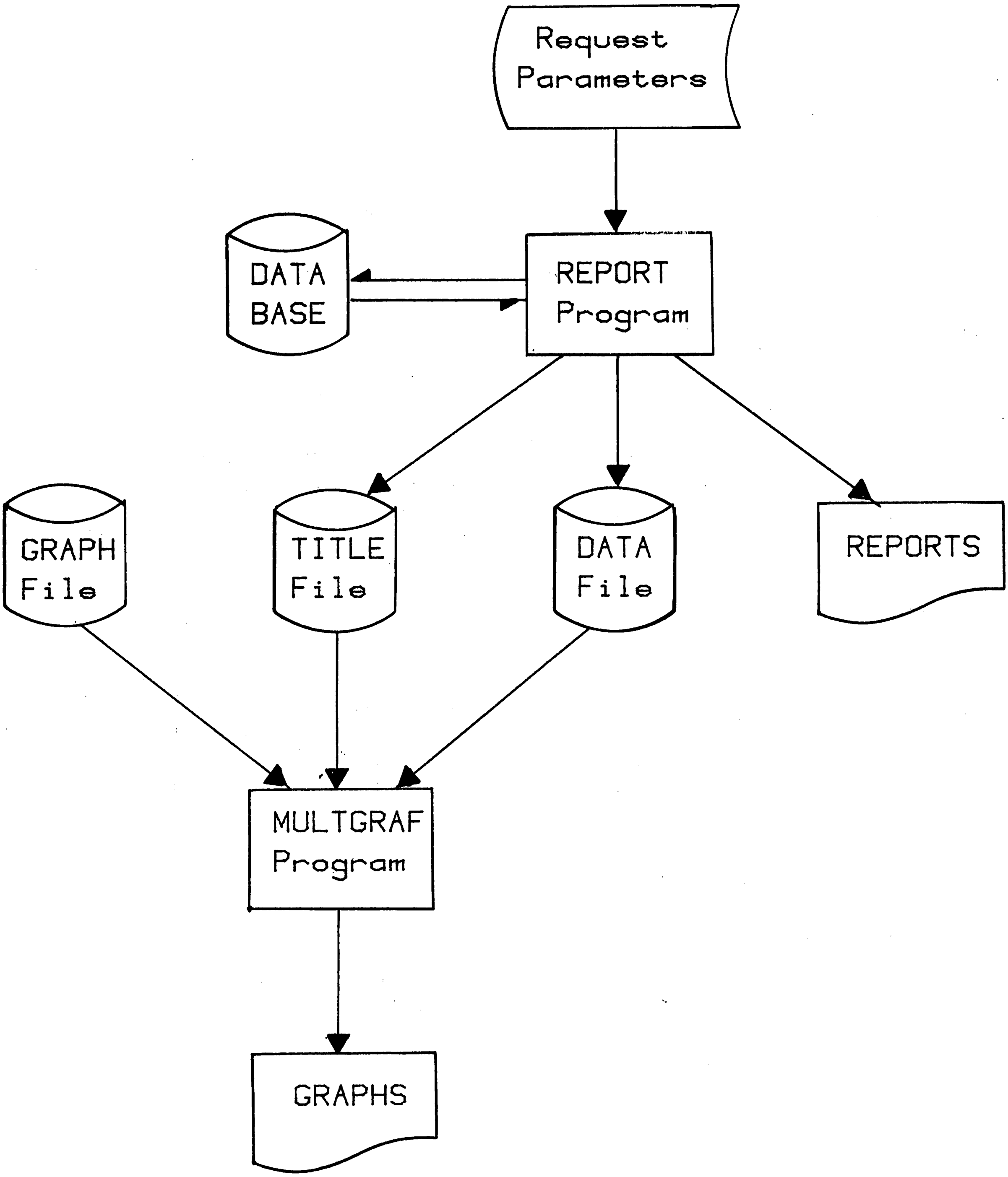

TERMINAL
DATA File
Tabulating Utility
Routed PRINT File
PRINT File
FLIP
DATA File
Format
SORT
TITLE File
GRAPH File
DATA File
MULTGRAF
GRAPHS

EXHIBIT 3


Request Parameters
REPORT Program
DATA BASE
GRAPH File
TITLE File
DATA File
REPORTS
MULTGRAF Program
GRAPHS