IMPROVING USER INTERACTION IN A BATCH ENVIRONMENT GRAPHICAL SYSTEM

James W. Wendorf* and Geoffrey N. Williams†
*University of Waterloo
†Atomic Energy of Canada Limited

ABSTRACT

Batch systems imply passive graphics. Active intervention while the batch program is creating the original drawings is not possible.

However, the user can play a very active role in the process of displaying the drawings if the drawings are saved in device-independent form and a simple but powerful graphical post processor is provided.

The particular problems of the user of a graphical post processor in a batch environment are discussed, and a set of basic concepts pertinent to the display of drawings on a variety of different graphical output devices is presented.

The design of an existing post processor based on these concepts is described, and reference is made to system dependent measures that can be taken to make the batch environment viewing process as convenient as possible.

AMÉLIORATION DE DIALOGUE DANS UN SYSTÈME GRAPHIQUE AVEC TRAITEMENT PAR LOTS

RÉSUMÉ

Les systèmes de traitement par lots ne peuvent fonctionner qu'avec des organes d'infographie passive, car il est impossible d'agir de façon active sur les représentations graphiques initiales pendant que le programme du traitement par lots les trace.

Toutefois, l'utilisateur a la possibilité de jouer un rôle actif lors de l'affichage des tracés si ceux-ci sont emmagasinés dans un support non tributaire de l'appareil et s'il dispose d'un post-processeur infographique simple et versatile.

On analyse ici les problèmes particuliers que rencontre l'utilisateur d'un post-processeur infographique dans un traitement par lots et on expose un ensemble de notions de base pertinentes à l'affichage des tracés sur une certaine gamme d'organes d'infographie.

La conception d'un post-processeur existant réalisé à partir de ces notions est décrite; de plus, on indique des mesures fonction du système que l'on peut prendre pour faciliter le plus possible l'utilisation des renseignements infographiques délivrés par un système de traitement par lots.

IMPROVING USER INTERACTION IN A BATCH ENVIRONMENT GRAPHICAL SYSTEM

James W. Wendorf
University of Waterloo
Waterloo, Ontario

and

Geoffrey N. Williams
Atomic Energy of Canada Limited
Chalk River, Ontario

Introduction

Many batch systems provide graphical output subroutines which users can call to produce drawings. Batch systems imply passive graphics since active intervention while the batch program is executing is not possible. However, a user can play a very active role in the process of displaying the drawings.

This paper discusses some features that can be provided in a batch system to assist the user in displaying the drawings he has created.

Methods of Handling Graphics in Batch Systems

Fig. 1 shows a common approach in which the user's program calls a graphics package, which produces a device dependent graphical file containing the instructions to produce a drawing on the device chosen by the user.

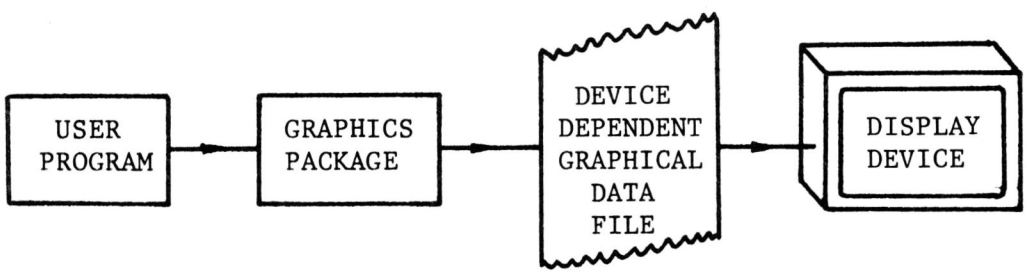

Fig. 1. A Common Method of Handling Batch Graphics

The use of a device independent intermediate graphical data file allows much more flexibility in the display process, as shown in Fig. 2.

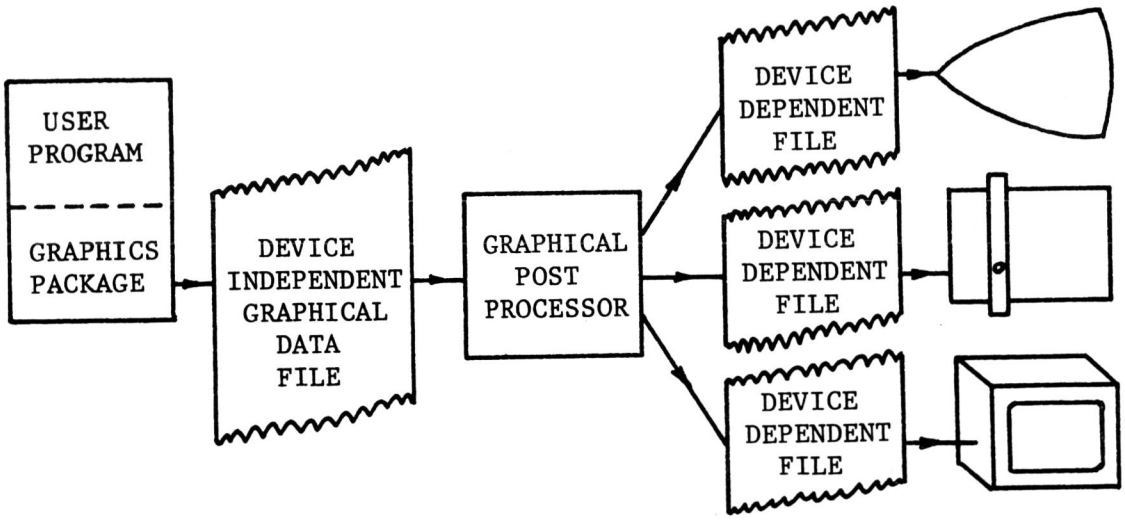

Fig. 2. A More Flexible Approach to Handling Batch Graphics

The graphical post processor is a batch job, but if the device independent file is saved, the user can run the post processor any number of times to produce various displays on any of the supported devices.

A number of systems in use today use this concept of a device independent file with a post processor (e.g. UNIPLOT from Control Data Corporation). However, it is crucial that these systems be designed to make the user's task as easy as possible. It is already difficult enough having to operate in a batch system.

The basic fact that the designer of the post processor must recognize is that the user of the post processor will, in general, have a very limited knowledge of the final appearance of a drawing created by his program, certainly much less than would be the case had he created it interactively.

Functions of the Post Processor

The post processor has two functions to perform. The first is a viewing function, to allow the user to view, on the device of his choice, a drawing created by his program. The second is a drawing manipulation function, to allow the user to create new displays from the original drawings or selected portions of the original drawings.

These functions can be provided in a number of ways. However, the limitations of operating in a batch environment must be kept in mind, and the post processor must be kept simple to be effective.

Capabilities Which Help Keep the Post Processor Simple

In the device independent graphical data file

(a) Store the graphical information in drawings which are given names DRAWING001, DRAWING002, etc.

(b) For each drawing define a DRAWING AREA, a rectangular space having physical dimensions, within which the drawing is located.

At the physical device level

(a) Give each device a name; e.g. TEK4015 for a Tektronix 4015 graphic display terminal.

(b) Define a PHYSICAL DISPLAY AREA for each device supported. This is defined as the physical area in which graphical information can be displayed on that device. For the device TEK4015 this would be an area 360 mm wide by 270 mm high.

Define a simple mapping from DRAWING AREA to PHYSICAL DISPLAY AREA as shown in Fig. 3. Select suitable defaults for parameters associated with the mapping, but allow users to specify values for these parameters. In particular

(a) Allow the user to define a DRAWING WINDOW in the DRAWING AREA associated with the original drawing.

(b) Allow the user to specify SCALE FACTORS which are applied to the DRAWING WINDOW to create a VIEWPORT.

(c) Allow the user to define an imaginary DISPLAY AREA and specify the location of the VIEWPORT within that area.

(d) Let the system map the imaginary DISPLAY AREA defined by the user onto the PHYSICAL DEVICE AREA of the device selected by the user, scaling the DISPLAY AREA down uniformly, if necessary, to ensure that all the information selected by the user is displayed.

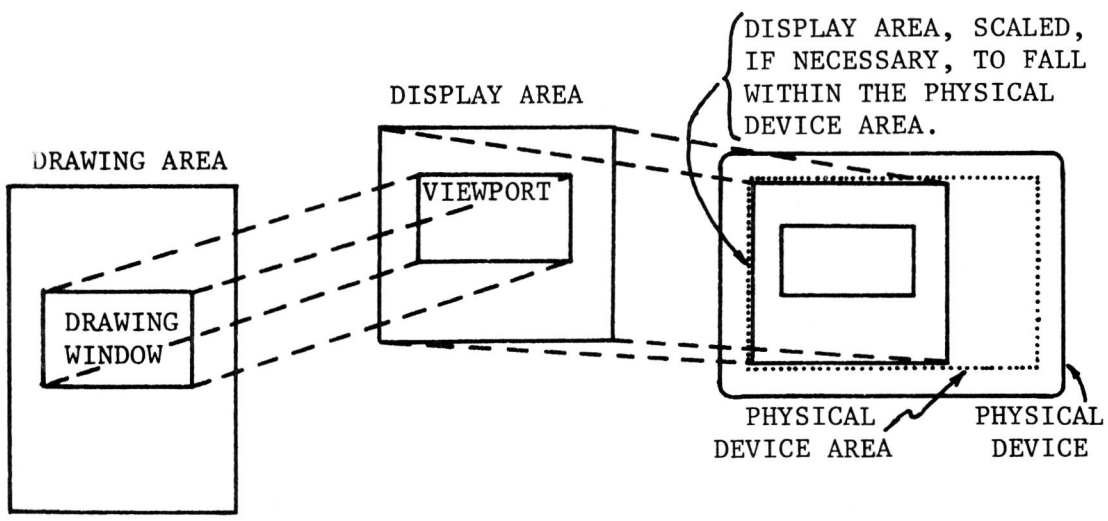

Fig. 3. Mapping the User's Drawing onto the Display Device

Note that

(a) The above capabilities allow simple scaling and stretching of selected portions of a drawing.

(b) All coordinates are in physical units, allowing easy visualization.

(c) Having the system map the imaginary DISPLAY AREA to PHYSICAL DEVICE AREA relieves the user of the burden of knowing the physical dimensions of the display surface, whilst giving maximum information by default.

(d) Allowing the user to set character quality is a useful feature.

Capabilities Which Make the Post Processor Easy to Use

A clear definition of the files used by the graphical post processor, as shown in Fig. 4.

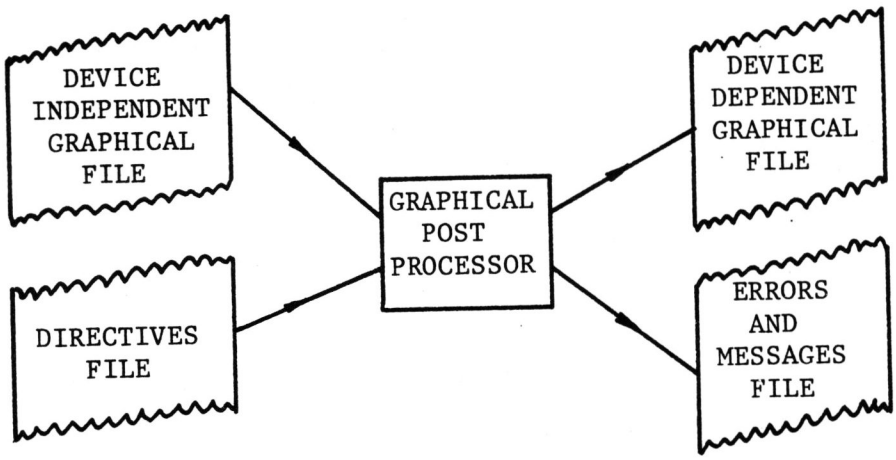

Fig. 4. Files Used by the Post Processor

Convenient parameter specification:

(a) Allow file and device type specification on the job control statement, and choose suitable defaults. The control statement

GRVIEW(D=TEK4015,GID=6P,I=0,L=0)

would cause the graphical output viewing program GRVIEW to execute, producing a device dependent file for a Tektronix 4015 terminal and routing that file to a terminal with the identifier 6P. I=0 and L=0 indicate that no input directives are provided and no print output listings are required.

(b) Provide "free format" directives of the form KEYWORD=value for the user to set values for parameters; e.g.

SCALEX=1.5,SCALEY=.75,CHQUAL=HIGH

(c) Have the keyword DRAW, a verb, cause the drawings specified in the rest of the directive to be processed for display according to the current settings of the display parameters; e.g.

DRAW=6,12-END,1-3,ALL

In general, repeat the cycle, resetting values and DRAWing until all directives have been processed.

Making the Batch Environment Iterative
Viewing Process as Convenient as Possible

The concepts and ideas presented to this point have been system independent. However, the viewing process should be made as convenient as possible by taking advantage of any existing system dependent features, and where justified, by providing new features.

For example, in the batch environment graphics system supported at the Chalk River Nuclear Laboratories, the following system dependent features are provided for the user.

(a) Each display has a display number added to it by the system for reference purposes.

(b) In response to the directive GRIDON=YES, the system will create a reference grid which can be overlayed on top of a drawing to provide the user with information about the physical dimensions of his drawing. Whenever a grid is requested, a list of the parameters that were used to create the display is also output.

(c) The information, i.e. parameters, used to create a display are saved in the device independent file. This allows displays that were viewed on one device to be output on other devices. This allows hard copy capability, for example.

(d) A system dependent overlay capability allows a drawing to be viewed first before a grid is overlayed on top.

(e) With an intelligent front end to the batch system a "pseudo-interactive" mode of terminal operation can prompt the user for the parameters to be used to create a display.

The first three features listed above are provided by the post processor which runs as a batch job and which makes extensive use of a graphical data management system developed at CRNL[1].

Conclusion

A number of basic graphical output requirements such as previewing drawings and obtaining hard copy of selected drawings can be quite adequately handled in a batch environment if the system supporting these capabilities is easy to use.

A useful level of interaction can often be provided by taking advantage of existing system dependent features without having to provide full interactive capability.

References

[1] J.W.WENDORF, "A Simply Extended and Modified Batch Environment Graphical System (SEMBEGS)", Comm. ACM 21, 11 (November 1978), 897-904.