EXPERIENCES WITH A SATELLITE GRAPHICS SYSTEM

C.H.M. Griffiths
Bell-Northern Research
Ottawa Canada

Abstract

Electrical Engineers & designers at Bell-Northern Research are engaged in design work in a number of different fields. The design process typically involves two-dimensional linear artwork. A variety of graphics equipment has been installed to aid the designers in this work, including digitisers, plotters and a number of CRT displays. Many of these devices are operated on-line to our central computer as time-sharing terminals.

This paper describes one of our system software developments. The IDIIOM graphics display system is a large refresh CRT system with a function keyboard and lightpen. This system is supported as a fully interactive time-sharing terminal, interfaced as a satellite graphics processor. The nature of our graphics support is critically described, as well as the program design, before considering the place of such systems in computer graphics in the future.

Résumé

Les ingénieurs électriciens et les concepteurs des Recherches Bell-Northern effectuent des travaux de conception dans divers domaines. Le Processus de conception comprend ordinairement l'exécution de nombreux dessins linéaires bidimensionnels. Divers appareils graphiques ont été installés pour aider les concepteurs dans ce travail, entre autres, des convertisseurs numériques, des traceurs et un certain nombre d'affichages cathodiques. La plupart de ces dispositifs sont reliés en direct à l'ordinateur central de la compagnie, comme terminaux fonctionnant en partage du temps.

Ce mémoire décrit l'une de nos réalisations en software. L'affichage graphique IDIIOM consiste en un grand système cathodique doté d'un clavier et d'un stylo électronique. Terminal en partage du temps entièrement interactif, le système sert de processeur graphique auxiliaire. Le présent mémoire donne une description critique de ce système et de la conception du programme, puis étudie le rôle que de tels dispositifs pourraient être appelés à jouer dans le domaine des applications graphiques de l'ordinateur.

The Graphic Environment.

Bell-Northern Research is involved in research and development into telephony, telecommunications and related fields. A significant proportion of this work involves the design of two-dimensional geometric patterns, for example the drawing of electrical schematics and the artwork for both integrated circuits and printed circuit boards. A large volume of graphic data is also produced in the form of program flow charts.

To aid our engineers and designers a variety of computer graphic devices has been installed. This equipment includes digitizers, plotters, photoplotters, storage tube CRTs and a satellite graphics system based on a refresh tube CRT.

These devices all access our central host computer. This machine, a large IBM System/360 Model 67, runs exclusively in a time-shared mode supporting typically 50-60 terminal users simultaneously. The graphics equipment operate variously off-line, on-line and through a combination of the two. One of our digitizers for example works off-line producing a card deck which can be entered at the host, while another produces a disk file off-line which can be transmitted over a communications link to the central computer. Similarly one of our plotters is driven directly on-line, while the photoplotter is driven from the card deck mentioned above.

It is our policy to provide full and flexible interchange between the devices at our engineers disposal. It should be possible to use the most convenient or suitable graphic aid for each particular step of a design process. It is also our policy to encourage use of the central computer for simulation and data analyses. One critical process for example is the preparation of data from digitized schematics and IC designs that will enable manufacture, testing and maintenance over the life of the electronic product, a span of 10-20 years after its initial development.

These imply rigorous data control. We are achieving this through standard data representation independent of the graphic device on which the data was created or is to be modified. Data translators are required to provide transfer between devices, but this effort is minimised by translation to and from a common central representation. A minimum number of translators are required by this approach, though a double translation is in some cases required for the transfer.

The Software Situation.

The graphics device user who elects to use a number of different devices is faced with a compatability problem: no two manufacturers have agreed on a common software interface and even the same manufacturer has different interfaces for different devices. The only support consistently offered is FORTRAN callable subroutines. These typically support the manufacturers hardware and offer but limited logical facilities. The user is left the task of intelligibly interfacing the device to his system, before he can even start on his application progam. This is a greater problem if he prefers to use a language other than FORTRAN!

This situation is understandable. To date the graphics market has been fairly restricted in terms of its size and penetration of the computer market yet the range of applications to which graphics devices are being put is considerable. FORTRAN callable subroutines are at the highest level at which support is worth encoding. However our requirements to support a wide and increasingly diverse range of graphic devices and to be able freely and easily to interchange data between them, mitigated further against use of supplied software in favour of our own develoment.

Graphic Terminals.

Let us turn our attention to graphic terminals and the characteristics of the one that is the subject of this paper. A graphic terminal is defined as a device with a CRT of some type, a typewriter keyboard for character input and some species of coordinate input device, the whole connected via a telephone link or channel to a host shared remote computer. One may further categorise such terminals as in figure 1.

A number of Simple I/O devices such as the Tektronix 4013, ARDS and Computek 300 are in use. These are Direct View Storage Tube (DVST) devices and are proving very popular. Storage tubes are relatively cheap, but suffer a number of graphic disadvantages. The refresh time is long since they are dependent on line speeds. Conversely, they can display a lot of data without flickering, a problem with refresh CRTs. Selective erasure is not possible. This is a sine qua non for dynamic displays and for interactive editting. Lithocon storage tubes offer this capability but do not yet have the resolution characteristics required for fine graphic work.

The IDIIOM System Hardware.

The IDIIOM system that has been installed at BNR offers both Buffered I/O facilities and those of an intelligent terminal. The IDIIOM consists of a large refresh CRT, a display generator (DG) which shares 32K of 16 bit words of memory with a Varian 620/f mini-computer. Local storage consists of a 3 million character disk storage drive. The CPU program builds a sequence of display orders in memory, that the display generator then interprets as a display program, to present the display on the screen. Interrupts, from a light pen, a function keyboard or the display program itself, supplied through the display generator enable dynamic and interactive modification by the CPU program of the display.

The IDIIOM shows buffered I/O characteristics when translating display data received from the shared host, buffer by buffer into an IDIIOM display program. This may be termed a passive function. It shows intelligent terminal characteristics when, using the light pen and the function keyboard the user is able interactively to influence the display view. This interaction is wholly whithin the IDIIOM terminal and requires no response from the remote shared computer.

GRIFFON.

GRIFFON is the name of the software we have developed for the IDIIOM. It has no meaning as an acronym (though we tried!), but as in the myth it bears significance for the combination of mutually incompatible functions that it encompasses. The title even, misspelt, has been retroactively applied as the software has evolved.

A key factor in the design of GRIFFON as mentioned above has been interchange between graphic devices through compatible representation on the mainframe. This has led to two major subsystems and the supporting of the IDIIOM as an interactive time-sharing terminal.

Communication with the 67 is over either of two communication links. Bulk data transfer is provided over a 4800 baud line while time-sharing interaction is over a 300 baud teletype line. This operates through a slightly modified teletype port on the 620.

Users have access to the full complement of time shared services and thus are able to interact with the mainframe independent of the IDIIOM display for the preparation and modification of data needed or used in a subsequent display operation. So far as software development is concerned, all our work has been done using normal keyboard terminals, with transfer of completed programs over the line to the 620. Without these links the progress we have made would not have been made.

The software required to handle the telephone links and to communicate with the time-sharing software, CP/CMS, was the first thing we tackled. This is now embodied in the first two GRIFFON subsystems, LOGIN and CMS. The LOGIN subsystem controls the users connection and initiation into CP/CMS, similarly to the way he would login at any other terminal. We ensure transparently that the high speed line is logically made available to him, enabling the various service programs he will run to operate without further setup operations. Once through LOGIN the CMS subsystem is invoked which enables full interaction with CMS and selection as the user wishes of the next subsystem to be used.

CALCID is the first major subsystem. We run a CALCOMP plotter driven by data held in disk files called PLOT files. CALCID provides a quick-view capability of such CALCOMP PLOT files at the IDIIOM enabling significant savings of plotter time through elimination of bad plots. CALCID is essentially a passive environment in that there is no facility through graphic interaction to alter the data displayed. The view of the data that is displayed can be changed and thus the validity of the graphic data confirmed, or errors discovered. Once satisfied the user can then commit the PLOT file to hard copy; or he can through CMS recreate his PLOT file till satisfied.

The key features built into CALCID are those enabling rescaling and relocation of the display view. These use the refresh speed of the display and its dynamic modification to full effect. Integrated circuits for example may contain 10-20,000 vectors and the designer needs to be able rapidly to WINDOW in to view the detailed arrangement of 100 or so vectors. A GRID may be overlaid to confirm vector alignment and displacement.

These features have proven especially useful when confirming the accuaracy of SP1 Flowcharts. SP1 is a BNR development and now product, of a telephone exchange that operates under Stored Program (or SP) control. Data describing the programs has to reach the customer running an SP1 telephone exchange, in as accurate a form as possible. The ZOOM, SLEW and MOVE commands of CALCID enable the programmer to follow through the logic of his program as recorded in the flowchart and highlights the errors that need correcting before producing the final hard copy. The TEXT and PAGE commands were provided for flowchart viewing. The TEXT command makes textual data visible, while the PAGE command selects the next section of the flowchart for examination.

A subset of the CALCID subsystem commands are provided in the second major GRIFFON subsystem, Grapple. Grapple was developed as an interactive graphic description language. As well as solving our data interchange problem, it is also proving itself as a graphic environment programming language. Our first implementation under GRIFFON is similar to CALCID in that the intelligence of the IDIIOM is not used interactively to modify the displayed data. However, the data can be interactively modified through the Grapple processor running on the 67. Local commands at the IDIIOM enable alteration of the view to be made several times faster than through interaction with the 67.

We have made no particular provision for those of us who are left-handed. Figure 5 shows the format of the IDIIOM as seen by a user. The lightpen and user provided menu lie within reach of the right hand. The location of the function keyboard and typewriter keyboard are not however fixed and may be relocated to suit either handedness. A five line scroll for textual communication is found at the bottom of the screen nicely at viewing height and within normal eye movements from the keyboards. This we believe provides a very convenient operating environment.

Structurally, GRIFFON has evolved into a program overlay arrangement. Storage restrictions led us to adopt this format originally, though once adopted of course, it gave us an open ended structure to which additional subsystems could be easily added. A restriction to the overlay structure which at the time of writing we are coding to overcome prohibits us from structuring multi-leg overlays: each of our subsystems is a single overlay.

The storage restriction arises because of a need to dedicate as much core as possible to the display file. A mask of 10-20,000 vectors requires 10-20,000 words of display file at least and maybe more when scaled to its maximum extent. This may be considered an unnecessary restriction. The IDIIOM hardware after all includes a feature that enables the program to determine whether a graphic entity is visible or invisible. In theory it is only necessary to maintain those entities which are at least partially visible in core, in an active display file. Invisible sections of the display file could be swapped back in from disk as the user MOVEs or SLEWs to bring them into view. By maintaining the whole display file in core, we implemented relocation with a minimum of effort.

It is worth noting that we do not need to do software scissoring. Perhaps if we did we would have developed a swapping algorithm and avoided storage restrictions. Scissoring is necessary on devices where "wrap around" can occur, i.e. where the display addressing scheme is limited to the viewing area. The IDIIOM is equipped with a "virtual" display area 4096 times the visible area, provided by 16-bit coordinate addressing. This gives a more than adequate limit to magnification, such that scissoring was unnecessary.

Scissoring from another point of view helps to reduce "flicker". Flicker, is a problem with refresh displays, arising when the volume of data displayed is such that the display generator cannot refresh the image within the decay time of the phosphor on the screen. This is not in practice too great a problem. Again, the IDIIOM hardware includes features that minimise flicker, by reducing display generator processing time when the displayed data is invisible. Flicker has only been a critical nuisance when viewing an image containing large amounts of data at a scale such that all the data was visible. Some masks pulsated psychedelically. At these scales though detail is hard to differentiate and productive use of the facility is made only after ZOOMing to a reasonable scale a scale at which flicker is in any case minimal.

The GRIFFON's Future.

Lewis Carroll's Gryphon wanted the adventure first and no explanations. In the same way our GRIFFON is likely to evolve rather than be tightly specified and planned. The key shortcoming in a graphic sense, is the absence of any real intelligence built into the IDIIOM, the absence in the code so far of interactive data manipulation. Evolution involves expediences and compromises. GRIFFON's evolution to date has been subject to the requirements of existing data structures in the 67. We anticipate evolving more interactive graphic functions local to the IDIIOM system.

The implementation of a GRIT interpreter at the IDIIOM is the third major subsystem to be implemented under GRIFFON, and falls within this last category.
Grapple, the interactive graphics language mentioned earlier, is implemented in a compiler, which like all compilers produces object text. For a terminal session, interactive with the 67, this object text remains in main memory, and is interpreted there producing the programmed visual effects at the specified graphic device. A Grapple feature permits the saving of the compiled text as a file on disk known as a GRIT file. GRIT files will be transferred to the IDIIOM and the interpretation done there.

The expediency for doing this is essentially one of speed. Time-sharing interaction with a shared central processor is not as fast as one would like, particularly for simple graphical functions like digitising and moving elements around. By coding an interpreter into the IDIIOM we will run at CPU speeds rather than transmission speeds providing a more responsive and far smoother graphic operating environment. At the same time the power of the 67 is retained for compilation. This has the further advantage of retaining data compatability, and hence of ease of transfer to other graphic devices.

Interactive Graphics at the sort of device described here is a sophisticated tool. Without question it is these sophistications that make such graphics attractive. In many cases though, the graphic facilities available through less intelligent terminals such as storage tube terminals, are more than adequate to the applications need. The intelligent terminal, the satellite graphics system, has the key advantage of speed. This advantage is meaningful only when used intelligently to its full extent, to do those things that are slow or impracticable at simple or buffered I/O graphic terminals. We anticipate more generally available hardware aids, such as 3-D rotation, hidden-line removal and sectioning. Whatever happens, satellite graphics systems will continue to have a valued place in our range of graphics equipment.

Graphics Terminals

Type	Characteristics
Simple I/O	All functions on host shared computer. Display directly produced the data received.
Buffered I/O	Local translation needed to produce display. Some data manipulation functions locally. Other processing functions supplied by host.
Intelligent Terminal	Graphics functions locally. Data storage and processing centrally.

Figure 1.

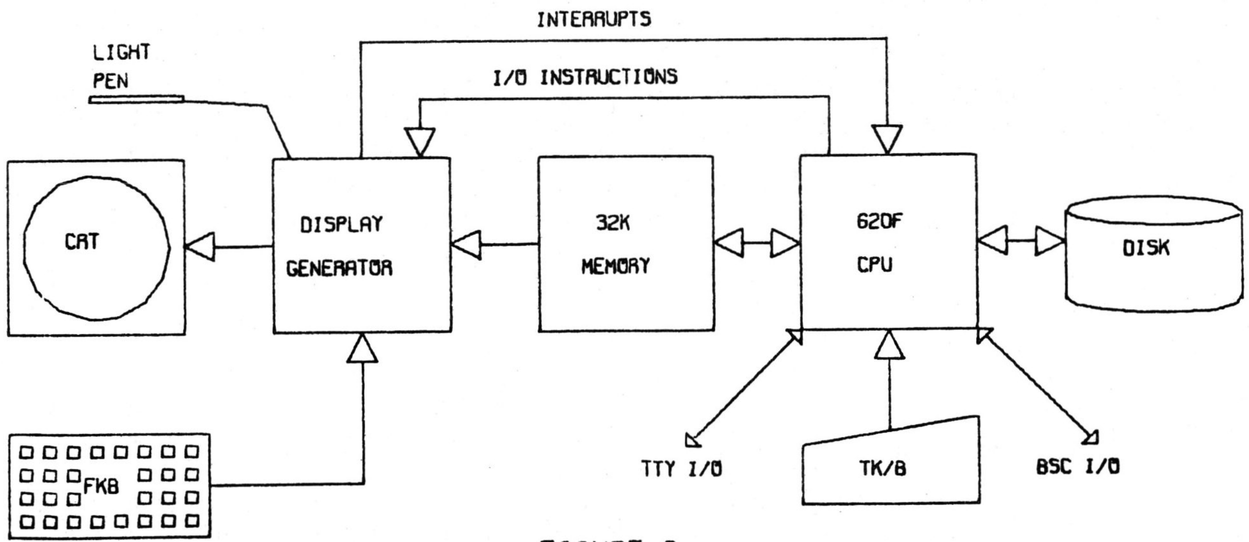

FIGURE 2

GRIFFON Subsystems

- LOGIN
- CMS
- CALCID
- GRAPPLE
- Maintenance
- GRIT

Figure 3.

CALCID Subsystem Commands

Control
    SELECT
    ERASE
    CATALOG

Scaling
    ZOOM UP
    ZOOM DOWN
    WINDOW
    RESET

Relative
    MOVE UP|DOWN|LEFT|RIGHT
    SLEW UP|DOWN|LEFT|RIGHT
    ROTATE

Miscellaneous
    TEXT
    PAGE
    GRID ON|OFF
    GRID SIZE UP|DOWN
    GRID OFFSET X|Y

Figure 4.

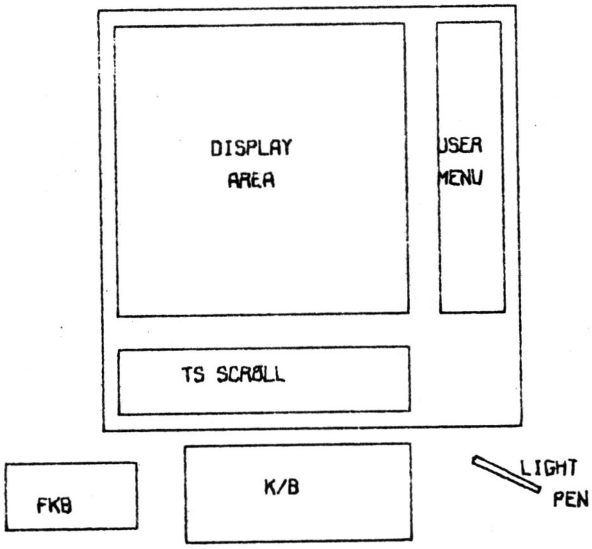

FIGURE 5

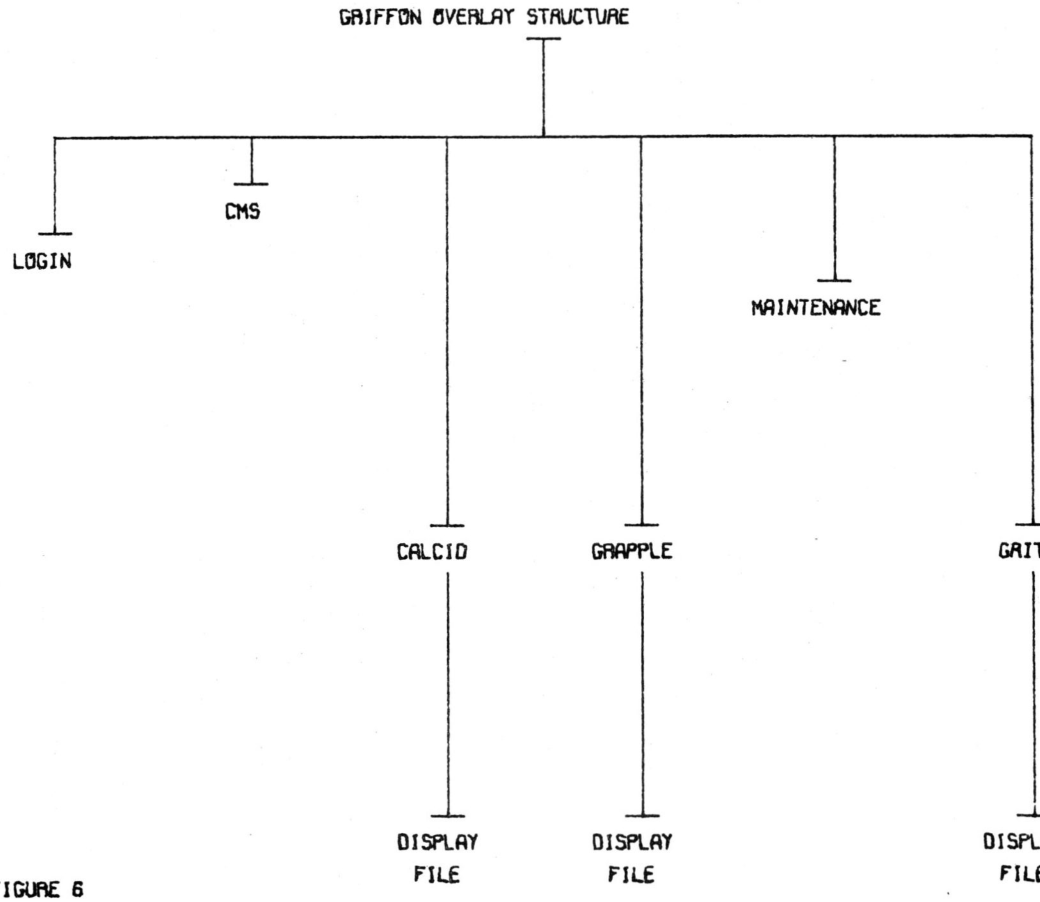

FIGURE 6