DEVELOPING GRAPHICAL TOOLS WITH MINIMAL GKS

Robert Sabourin
Software Development, Infolytica Corporation
Montreal, Quebec
D. A. Lowther
CAD Laboratory, Dept. Electrical Engineering, McGill University
Montreal, Quebec

ABSTRACT

The Graphical Kernel System (GKS) is emerging as an important standard for the display and manipulation of two dimensional data. Tools described in this paper were developed for the minimal GKS implementation which has both input and output capabilities. Minimal GKS (level MB) is guaranteed to exist in any GKS implementation with both input and output capabilities[1].

Minimal GKS provides sufficient power to build complex menu driven applications programs while maintaining independence from the graphical hardware used. Info-Menu is a tool for creating interactive menu driven interfaces. Info-Graph is a library of subroutines which enables users to create line graphs with various display attributes and axis mapping functions. The implementation of both packages will be discussed.

KEYWORDS: Graphical Kernel System, Graphics Standards, Graphical Tools

INTRODUCTION

Two graphical tools developed at Infolytica Corporation for use in the CAD system MagNet will be described. The Info-Menu package provides a set of Fortran 77 subroutines which allow for the creation and manipulation of Graphical menus. Menu selection is provided by a locator input device. The Info-Graph package is used to create 2-D line graphs allowing user control of axis mapping functions and of the GKS attributes which will appear in the graph. Both utilities use only the functionality of minimal GKS.

GRAPHICAL TOOLS

The graphical toolkit is a concept similar to, and is part of, the programmer's toolkit. The latter consists of compilers, linkers, debuggers, text processing programs etc. The graphical toolkit consists of the basic building blocks required to create computer graphics based applications programs.

Tools exist today which automatically create menu driven user interfaces [2], as well as providing libraries of subroutines for graphical output or input. Most of these tools depend on certain types of hardware, computers or operating environments. Even the most commonly used tools (such as Plot-10 by Tektronix Inc*) are bound to a certain class of graphics devices. The tools discussed here use only minimal GKS and are implemented in Fortran 77 so as to be portable across many different graphics devices and computers. Info-Menu and Info-Graph are implemented on HP-9000 series 200 and 500 mini-computers, various Digital Equipment Vax computers running both VMS and BSD 4.2 Unix and various 68000 based machines. Making the tools available on new devices merely involves linking them to a GKS which supports the device. Applications software requires no code changes except possibly to specify the new GKS workstation type.

The Info-Menu and Info-Graph packages are designed to allow the user to create interactive, device independent applications programs. Figure 1 illustrates the position of these tools below the applications software.

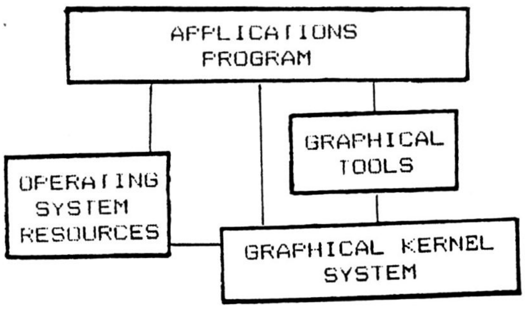

Figure 1. Graphical tools position relative to GKS in an applications program.

THE GRAPHICAL KERNEL SYSTEM (GKS)

GKS is an international graphics standard for 2-d applications. A 3-d extension of GKS is being prepared at this time [3]. GKS defines a set of language independent structures and procedures which are bound to programming languages as subroutines or statements. The capabilities of GKS include the basic output primitives of polylines, polymarkers, fill areas and text strings. Logical input in GKS includes locator, choice, string, stroke, valuator and pick. GKS allows applications to access multiple active workstations. Attributes (such as line or marker types) can be controlled through global or workstation dependent attributes[4][5].

GKS was designed to address a rich set of graphics operations. Since many devices exist which cannot support all the features of GKS, the standard allows GKS implementations of various levels. Each level of GKS has a well defined set of functions. Any valid GKS implementation must conform to one of the standard levels of GKS. Three levels of logical input capabilities and four levels of graphical output capabilities are defined. The input levels are specified as A,B and C whilst the output levels are M,0,1, and 2. An implementation has an output and input level, for example MA, 2C, etc. The following is a brief description of the levels.

Output levels

Level M: GKS has basic output primitives with no workstation dependent bundled attributes.

Level 0: includes all of the functionality of level M with the ability to bundle attributes such as line types and character fonts to workstations. A global index may refer to different attributes for each workstation.

Level 1: includes all of level 0 with segment storage and metafile input and output.

Level 2: includes all of level 1 with the addition of workstation independent segment storage.

Input levels

Level A: provides no graphical input.

Level B: includes request input for locator, stroke, string, choice, valuator and pick input.

Level C: includes all of level B with the addition of sample and event driven graphical input.

In the context of this paper minimal GKS is the level MB. This level is significant in that it addresses the widest selection of available devices currently used in interactive graphics.

GKS COORDINATE SYSTEMS

GKS uses three coordinate systems, world coordinates (WC), normalized device coordinates (NDC) and device coordinates (DC).All output and input position information is in terms of a user defined world coordinate system. The extent of this system is set by the applications program as required. GKS transforms world coordinates into a normalized space with a range of 0 to 1 on both axes. The mapping from world to normalized coordinates is via a user selected normalization transformation. Each normalization transformation is specified by the GKS user as a rectangular window in world coordinates mapped to a rectangular viewport in normalized device coordinates.

Clipping to the viewport of the normalization transformation may be enabled or disabled by the user. Each workstation in a GKS implementation has a workstation transformation mapping part of NDC to some part of the device coordinate system.

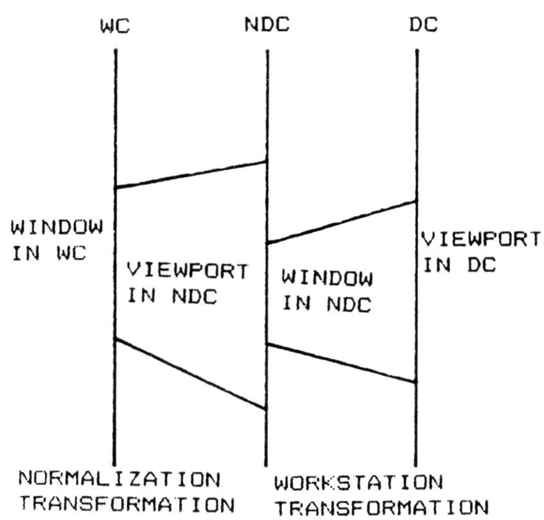

Figure 2. Normalization and workstation transformations.

Since many normalization transformations may have been used to display data, locator and stroke input always return world coordinates together with the normalization transformation, whose inverse was used to obtain them. When the viewport of more than one normalization transformation contains the coordinate selected, the transformation with the highest VIEWPORT INPUT PRIORITY containing the point is selected.

INTERACTIVE MENU COMMUNICATION WITH MINIMAL GKS

A typical user interface to CAD/CAM programs accepts commands in two ways. The user may (1) communicate via a dialogue with the applications program or (2) enter a command as a set of menu selections by pointing to a graphical display surface using a locator input device such as a tablet and stylus, a joy stick , a mouse, etc[6]. Locator input devices may be used for menu selection and locator input simultaneously. For example, when a user enters point coordinates, locator input may trigger a menu selection or coordinate data depending on the position of a cursor on the display surface. If the cursor is positioned over the menu, and locator input is triggered, the input is assumed to be a menu selection. If the cursor is positioned over the working area, and locator input is triggered, a coordinate value is interpreted by the applications program.

The Info-Menu package is a set of Fortran 77 subroutines callable by applications programs[7]. The primary raison d'être of the Info-Menu package is to provide both menu selections and locator input to an applications programmer through a common software interface. The ability to use the same logical input device for both locator and choice requests can be implemented in a level MC GKS for the case of a locator device triggered by a choice selection. For example, a graphics terminal with a joy disk can be used. To trigger locator input the user strikes a character thus indicating a choice. Many applications developed to run on a Tektronix 4010 style terminal use interactions of this type. In order to emulate the required function, the program would have to perform the following:

REQUEST LOCATOR returning the position
SAMPLE CHOICE returning the selection.

The availability of input level C, however, is rare, owing to the operating system services required to implement asynchronous event driven input.

INFO-MENU

The GKS environment in which Info-Menu is running is determined by calling GKS inquiry functions. No attempt to use a function, or control an attribute, is made without verifying that this operation will not cause a GKS error. The following state variables of GKS are inquired before Info-Menu will initialize:

Inquire GKS operating state:
-must be open

Inquire level of GKS:
  -must be M or higher output
  -must be B or higher input

Inquire Workstation State:
  -must be open and active

Inquire Locator Device Information:
  -the specified locator device must exist

The applications program must allocate a list of normalization transformations to Info-Menu for displaying menus. For each transformation allocated to Info-Menu the user may have one visible menu. GKS normalization transformations should be considered as resources; controlled, allocated, and recovered by the applications software. Any utility package should use normalization transformations allocated by the applications program.

The state of GKS before and after calls to the Info-Menu package is the same. This is accomplished by saving any GKS attributes before changing them. For example, an Info-Menu subroutine exists for making a menu visible; when the subroutine displays text in a menu button it modifies most character and text attributes. To implement this the following must be done:

```
save text colour index
save character height
save character up vector
save text alignment

compute new height
set text colour index
  ( from menu definition )
set text up vector to (0.0,1.0)
set text alignment to (normal,normal)
   ( this alignment draws text from
       left to right )
set character height
draw text in menu buttons (centered)

restore text alignment
restore character up vector
restore character height
restore text colour index
```

All save operations are implemented with GKS inquiry functions. All set operations use GKS set attribute functions.

The use of menus in an applications program may be logically separated into three phases:

1- Definition of menus.
2- Setting display attributes of menus.
3- Interaction with menus.

The definition phase involves specification of menu contents to Info-Menu. Each menu is stored in the internal data areas of the package. The applications program may delete any menus when they are no longer required, thus freeing space for more menus.

A menu consists of a set of buttons, each containing a menu item. Menu items may be any basic GKS output primitive available in level M; they include: POLYLINES, POLYMARKERS, FILL AREAS and TEXT. A menu item may also be a solid rectangle containing a colour index. Solid rectangle menu items are included to facilitate colour selection menus. A subroutine is provided to create menu items of each basic type. Since an applications program may require the same items to appear in many menus the creation of menu items is separate from the creation of menus. A typical example is an applications program which requires each menu to include the string 'HELP' in its first button. In Info-Menu this item would only have to be created once. The item is assigned an integer identifier which is used for all subsequent references to it. The coordinate space of a menu item is from 0 to 1 in both directions. GKS primitives placed in menu items will always be clipped to this extent.

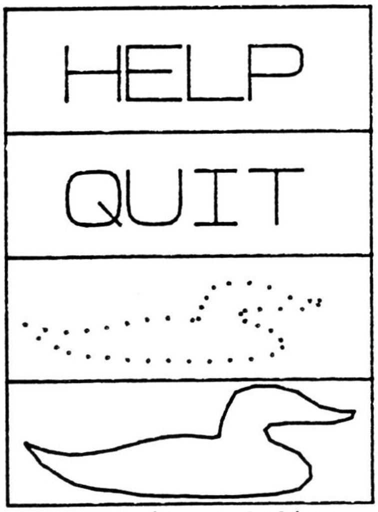

Figure 3. A menu with polyline, text and polymarker buttons.

Menus are created with the Create Menu routine. The applications program provides an integer array containing the list of items in the menu. The order of items specified will be the order in which they appear in the display. Menus are allocated identifiers by Info-Menu. Whenever a menu is selected, both the identifier and the button number are returned. The position and display attributes of a menu may be set by the applications program. Each menu is considered to have one button per item. The buttons may each have different colours and primitive attributes. The available primitive attributes are 'linetype' for polylines, 'markertype' for polymarkers, and 'interior style' for fill areas. GKS enumerated types are used to describe the necessary attributes.

When menus have been created, the user has a choice of various highlighting styles, colours and GKS primitive attributes to assign to each menu button. The position of the menu is specified by the applications program using GKS normalized device coordinates.

Info-Menu allows the applications program to control which menus are visible at any time. When a menu is visible, it is selectable. The applications program may set one of two interactivity modes :

1- Locator input or menu selection
2- Menu selection only

Menus may be made visible (or invisible) at any time. Info-Menu is allocated GKS normalization transformations when opened. The number of transformations allocated to Info-Menu determines the maximum number of menus which may be visible at any time. Each menu button is displayed in a coordinate space ranging from 0 to 1 in both directions. The display of menu items in the buttons is accomplished by setting the window (WC) of a normalization transformation to min=(0,0) max=(1,1) and by sliding the viewport down the menu.

A menu selection consists of:

1: A menu identifier (which menu did the user choose)

2: A button number (which choice was made)

3: A locator input coordinate pair (in world coordinates)

4: The GKS normalization transformation used to transform the device coordinate to a world coordinate

5: Logical status of returned data (is the input locator or menu selection)

6: Error information.

The request menu selection operation is implemented using normalization transformations. One transformation is assigned to each visible menu. The viewport of each transformation is set to the NDC menu position. The viewport input priority of each transformation is set higher than all other normalization transformations. Since menus do not overlap, the normalization transformation number returned with GKS locator input may be used to determine whether a menu or location was specified.

The world coordinates of each menu transformation are set so that the Y values indicate the button number. Menus are displayed such that button 1 is the highest on the display; all other buttons appear in order below button 1. The window specified for a menu with 6 buttons ranges from ymax=-1.0 to ymin=-7.0. The GKS locator input:

x=1.0, y=-5.3, nt=1

indicates that button 5 of the menu was selected.

INFO-GRAPH

The Info-Graph package allows users to create 2-d line graphs from ordered sets of data. This tool is a subroutine library which may be embedded in the application program. Minimal GKS with no input capabilities is a sufficient environment to use this package[8].

A graph must be allocated a position on the display. This is done by specifying an extent in normalized device coordinates in which the graph will appear. This position is divided into an inner and an outer rectangle. The user's function will be plotted in the inner rectangle. Descriptive text and the function extents are placed in the outer rectangle. The minimum text in an Info-Graph is: X axis label, Y axis label, Graph Title, function extent along X axis, and extent along Y axis.

The cartesian coordinate system in which the graph data resides is referred to as WS so as not to confuse it with GKS WC space. Info-Graph maps this data into normalized graph space (NGS). NGS has an extent of (-1,+1) along x and y. The mapping from WS to NGS is a scaling operation which uses separate scaling factors on each axis. The mapping fits all of the graph into NGS space. Axis mapping functions are defined in NGS. Before the graph is plotted it is transformed from NGS to Function Space (FS) using the Axis mapping functions. The extent of FS is used as the world coordinate window of the graph being plotted.

Graphs may be displayed as a polyline, a polymarker or both. Tickmarks may be equidistant in WS, specified by the number along each axis, or at fixed positions specified by the user. The axis is always displayed as a polyline. Tickmarks may be displayed as polymarkers or polylines. If polylines are used the effect is a grid. The extent of the graph along each axis is displayed as text in the outer rectangle of the graph. Descriptive text for each axis and for the graph itself may be specified by the user. Info-Graph can compute text position or allow the user to place the text where desired.

When Info-Graph is initialized a GKS normalization transformation must be allocated to it. This transformation will be used to display the graph. All GKS attributes which require modification during execution of Info-Graph subroutines are saved and restored.

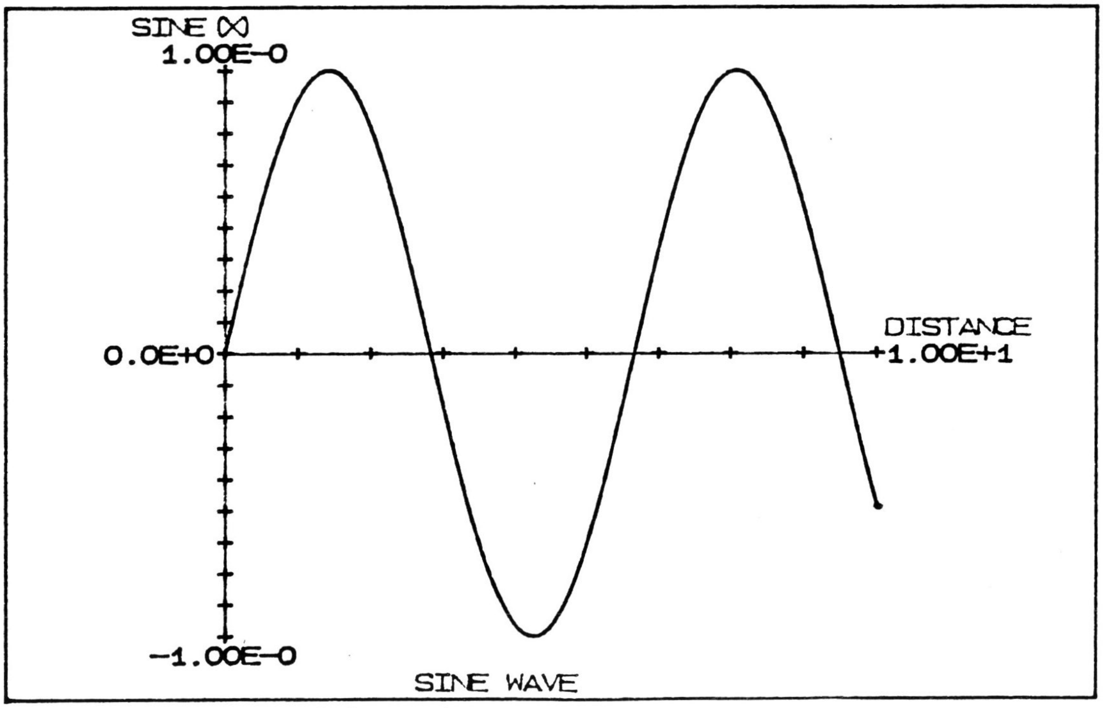

Figure 4. Default Text Positions in a graph. All text is displayed in the outer rectangle. The axes, tickmarks and data are displayed in the inner rectangle. Tickmarks are set to the GKS marker '+'. Data is displayed using a solid polyline.

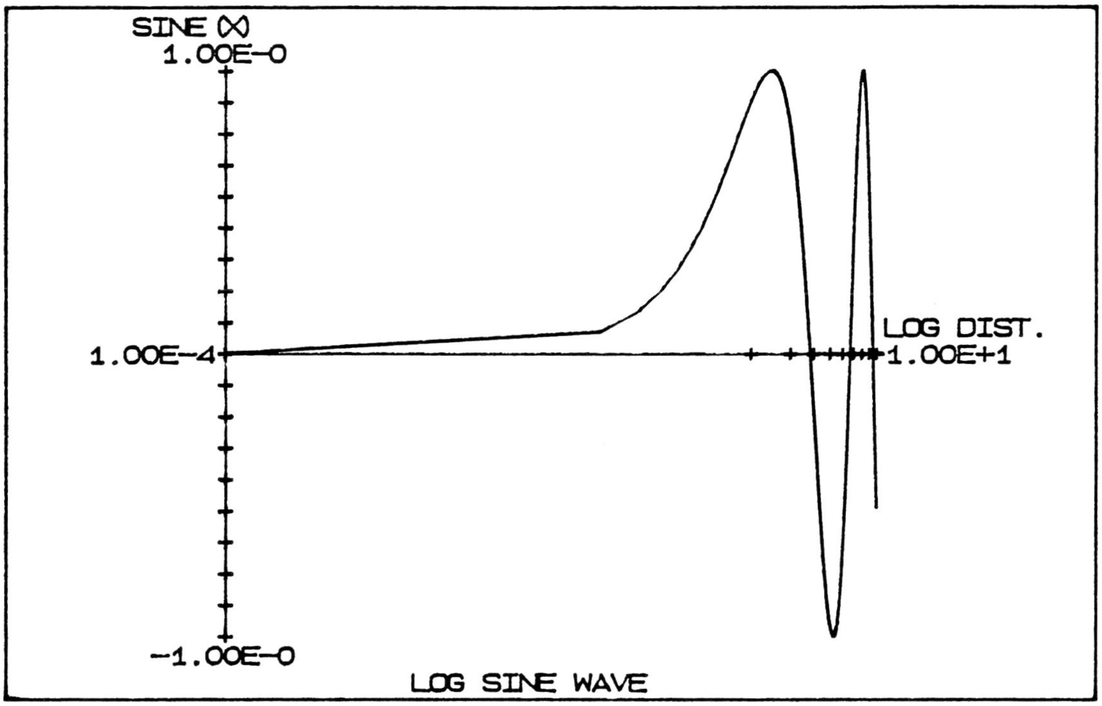

Figure 5. Axis mapping functions. The same data from figure 4 is displayed with a linear axis mapping function in the Y direction and a logarithmic mapping in the X direction. Note that tickmarks are equidistant in WS but not in FS.

The following mapping functions are made available to the Info-Graph user:

Function Number	Function Input	Function Output
1	all x	y = x (linear)
2	x <= 0 x > 0	y=0 y=log10(x)
3	x <= 0 x > 0	y=0 y=ln(x)
-1	all x	y = x (linear)
-2	all x	y = 10**x
-3	all x	y = e**x
0	user specified external function	

The GKS world coordinate space used to display the graph does not directly map to the extent of data being displayed, thus a facility to map coordinates in the GKS WC window to the Info-Graph WS extent is provided. The user may, at any time, find the correspondence between a WC point and its WS value. Info-Graph will also find the nearest point on the graph to coordinates specified in WS or WC.

CONCLUSIONS

GKS can be used to create graphical tools for developing interactive CAD/CAM packages. This paper illustrates that minimal GKS provides sufficient power to build complete menu driven applications programs. Info-Menu and Info-Graph were designed by Infolytica Corporation as stand alone GKS utilities. Currently all of Infolytica's CAD software uses Info-Menu for interactive communications. Info-Graph is extensively used in modelling and post processing. These tools have allowed Infolytica to port applications software to many new mini and micro-computers having varied graphics capabilities.

REFERENCES:

[1]American National Standards Institute: Information Processing Systems, Computer Graphics, Graphical Kernel System. X3H3/83-25r3, Jan. 1984.

[2]W. Buxton, M.R. Lamb, D. Sherman, K.C. Smith, 'Toward a Comprehensive User Interface Management System', Computer Graphics, Vol. 17, No. 3, July 1983.

[3]American National Standards Institute, GKS-3D Working Document, 1984.

[4]F.R.A. Hopgood, J.R. Gallop, D.C. Sutcliffe, D.A. Duce, 'Introduction to the Graphical Kernel System (GKS)', Academic Press, NY, 1983.

[5]G. Enderle, K. Kansy, G. Pfaff, 'Computer Graphics Programming GKS-The Graphics Standard', Springer-Verlag, Berlin, 1984.

[6]J.D. Foley, A. Van Dam, 'Fundamentals of Interactive Computer Graphics', Addison-Wesley, Massachusetts, 1982. pp 208-213.

[7]Info-Menu Users Manual, Infolytica Corporation, 1984.

[8]Info-Graph Design Notes, Infolytica Corporation, 1984.

* Plot-10 and TEKTRONIX are registered trademarks of Tektronix inc.