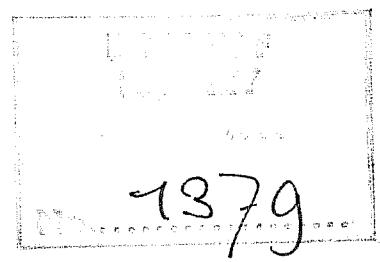
WORKSHOP

on

IMPLEMENTATION OF GKS AT ECMWF

14-15 May 1981



September 1981

REPORT ON THE WORKSHOP ON THE IMPLEMENTATION OF GKS AT ECMWF

by

H. WATKINS

Telecommunications and Graphics Section
ECMWF
Shinfield Park
READING
Berks.

Workshop Report
Operations Department May 1981

C O N T E N T S

List of Participants

The following is a list of those participating in the Workshop.

R. Buhtz	Free University of Berlin
A. Ducrot	INRIA
G-R. Hoffmann	ECMWF
A. Lea	ECMWF
A. Lemaire	ECMWF
J. Schönhut	Erlangen University
D. Söderman	ECMWF
H. Watkins	ECMWF

ECMWF Workshop on GKS Implementation

14 - 15 May 1981

AGENDA

Thursday 14 May

Morning: Informal discussions - presentation of ECMWF computing and graphical facilities.

Afternoon: Workshop on GKS implementation

- . implementation policy
- . Electrostatic plotter workstation
- . Vector to raster conversion - techniques and relationship with GKS
- . Device independence and the Metafile concept
- . Operational use of a Metafile
- . Adaptation of existing packages

Friday 15 May

Morning: Informal discussions

Afternoon: Seminars (see next page)

Seminars

The following Seminars were given on Friday 15th May as part of the normal ECMWF Seminar program. As such they were open to everybody.

A. Ducrot: "Current activity in the International review of GKS"

A. Lemaire: "Presentation of GKS. Expected evolution of ECMWF graphical software facilities"

J. Schönhut "Graphical software facilities at Erlangen University"

R. Buhtz: "GKS implementation at the Free University of Berlin"

1. Introduction

Following the Graphic section's reviews of the recent activity in Standards for Graphics, a decision was made to implement GKS ("Graphical Kernel System") at the Centre. Reference (1) gives a broad outline of GKS and why it was selected. Reference (2) describes the implementation policy that was adopted.

The ECMWF implementation was the result of a co-operation between the Centre's Graphical staff (A. Lemaire, H. Watkins) and A. Ducrot of the Institut National de Recherche en Informatique et Automatique, (INRIA), Rocquencourt, France.

At the time of the Workshop a preliminary implementation of GKS had been achieved. The purpose of the Workshop was to present our implementation policy and invite criticism.

Mr. Buhtz of the Free University of Berlin had been invited because of his work on vector to raster conversion techniques on a Cyber (Reference 4) and because of his work on a GKS implementation.

Mr. Schönhut of Erlangen University was invited because of his GKS implementation and because he has attended various meetings associated with the acceptance of GKS as a DIN standard.

Mr. Ducrot of INRIA was invited because he is a joint implementor of the Centre's GKS package, but also because he is Chairman of the French AFNOR (Association Francaise de Normalisation) Committee which is concerned with the acceptance of GKS as an international standard.

The Workshop took the form of a series of informal discussions which are summarised in the following pages under the headings outlined in the Agenda.

Associated with the Workshop was a series of Seminars on topics related to GKS and Graphical Software. These Seminars formed part of the normal ECMWF Seminar programme and are not reported here.

2. Implementation

2.1 Implementation policy at ECMWF

Alain Lemaire outlined the Centre's implementation policy. Although this has been described before (Reference 2) a summary is reproduced below for completeness.

A decision to use GKS was made after a review of standards in graphics. The only other contender was the GSPC core proposal, but that was rejected primarily because it is a 3-D system. Although meteorology deals with a 3-D world, the viewing transformations which are used cannot be performed by the CORE 3-D viewing system. Therefore to use a 3-D system entails unnecessary overhead.

Another reason for choosing GKS is that it is the only proposal currently accepted by ISO as a work item.

Since our applications are mainly batch (the Tektronic terminals are not used interactively but only for the display of pictures) we decided to restrict ourselves to a level 1 implementation of GKS. We planned to upgrade this later to a level 2a implementation purely because of the segment facilities which we decided to use for pre-defined backgrounds.
We decided to let the workstation do most of the work (e.g. clipping) and felt this followed the philosophy of GKS particularly in view of the trend towards more intelligent hardware.

This, however, means a departure from GKS(5,6) as we will perform the clipping only once at the device driver level. Normally in GKS one has 2 separate transformations and 2 separate clippings - firstly from World Co-ordinates (W.C.) to Normalised Device Co-ordinates (NDC), and secondly from NDC to Device Co-ordinates (DC).

Although the transformations can be combined, the clippings can only be combined if we do not have segments i.e. only in a level 1 implementation. The need to combine the transformations and clippings comes from a need to reduce CP time as much as possible.

Our GKS package is divided into 4 layers:

i) The GKS user interface - this passes across functions and data.

ii) The workstation manager - this sends commands to various workstations, performs the open -, activate - workstation.... etc.

iii) The DI/DD (device independent/device dependent) interface for all workstations.

iv) The workstations themselves.

The DI/DD interface is itself split into 4 sections:-

i) control and output

ii) inquire about output

iii) input

iv) inquire about input.

The main raison d'être behind this split is that we don't intend to use input, so we only need code i) and ii)!

For text we will define the dimensions in W.C. (which for us are inches as on the plotter), but we will store these in DC. Hence, a change of either window-to-viewport transformations, (WC - NDC or NDC - DC) will not affect the text sizes.

So far we have written all the user interfaces apart from a few inquiry functions. We have a debug workstation which also acts as a 'model' workstation for new devices. After experimenting with vector-to-raster conversion techniques we decided to utilise two Versatec workstations, one generating the raster file directly, the other utilising an intermediate vector file (see section 3 for further details).

A. Ducrot is writing a Tektronix workstation at INRIA. We have still to define the Metafile workstations, and we have decided

that for the Rutherford FR 80 COM, each camera will have a separate workstation.

It was originally planned to implement an Aydin workstation, but it now seems likely that we will send meteorological data to the Aydin and perform graphics locally. However, in order to send information back to the Cyber we need some sort of metafile.

As already mentioned, we intend, eventually, to upgrade to GKS level 2a in order to implement a feature which is very useful and heavily used in our current software - namely the predefined backgrounds. We would like to do this within the framework of GKS and after much discussion our final decision was via segments: hence the need to upgrade to level 2a. However, for this type of application of segments we do not need insert segment, transform segment or segment storage (see section 2.4 for further comments on this aspect of our implementation policy).

2.2 Implementation policy at Erlangen University

J. Schönhut described the GKS Implementation policy at Erlangen University. They decided to make a pilot implementation at GKS level 5.2. The emphasis was not to be on the speed of implementatic but rather on producing as professional an implementation as possibl This is especially important in the Erlangen environment which has a large spectrum of users on 10 different computers.

One problem concerned the location of segments. Interactive programs need a fast response which leads to a requirement of keeping the segments in core. On the other hand batch jobs do not need a fast response and segments need not be in core all the time.

This can be solved on a specific computer by using the computer manufacturer's extensions to standard Fortran, but becomes a problem when many machines need to be catered for. The solution adopted at Erlangen was to take a Pascal compiler and to extend the language to include "data defined on this medium". Definitions were also added to be able to choose the GKS level (1, 2a, 2b, 3 etc.).

The idea is that the output of this modified Pascal compiler will be a Fortran dialect specific for a specific machine.

The current status is that a GKS level 3 implementation is running on a Cyber with a test workstation, and "protocol testing" has been carried out. The compiler generates correct COMPASS code, and the first Fortran version is expected soon.

2.3 Implementation policy at the Free University of Berlin

The Implementation policy at the Free University of Berlin was explained by Buhtz. An implementation was started 1 year ago primarily for two reasons:

i) The existing software 'BIZEPS' works via a neutral metafile (i.e. a pseudo display file). The usage of graphics at Berlin was exploding not only in terms of new users but also new hardware. Furthermore there was an obvious requirement to have common software on all machines. However, the adaptation of BIZEPS to new hardware was not easy.

   A better solution was seen in the use of a GKS portable metafile rather than a low level pseudo display file, even though the latter might be more efficient once it is written.

ii) Since the linestyles, clipping and other operations are the same for a variety of workstations, a layered model was constructed. This fits well with the GKS philosophy whereby a workstation does not need to return control to GKS, but uses algorithms from a common pool.

A decision was made to use ANSI Fortran as much as possible, with UPDATE defines around machine specific code. However, even in this case code was written in ANSI Fortran to simulate the machine specific code (eg using 1 character/word for greatest portability).

In order to avoid large programs, workstation drivers are kept in capsules and are loaded via the fast dynamic loader. No OVERLAYS or SEGMENTS (in the Cyber loader sense) are used.

Up to now a level 3 implementation is running on a Cyber 172 and an IBM 370/115. The latter was the test site for portability, and the small size of user program is noteworthy (about 90K) especially since a GKS version from Darmstadt was about 3 times the size.

2.4 Discussion on implementation

There was a general consensus of agreement with our implementation policy. Some useful discussions took place regarding our proposed upgrade to level 2a at a later date.

It was explained that in the Centre's currently implemented software, a very useful feature was the predefined background, which could be recalled at will. It would be nice to introduce this via GKS, but how? Basically three methods had been considered:

i) via an escape function;

ii) via the natural GKS concept of segments;

iii) via some sort of metafile.

The final decision was to use the segment concept, even though this presented problems i.e. we don't require insert-segment, transform-segment or segment-storage. Furthermore the problem of combining clippings as mentioned before, made such a choice less than 100% desirable. It also meant an upgrade to level 2a.

It was agreed that although the segment was the approach that would appear to be closest to the GKS philosophy, an escape function could be used for a predefined background rather like a forms flash. There is an obvious need to prevent the proliferation of GKS dialects - which is what we would be doing if we used the segment concept. However, the escape function is documented as a non-portable feature and is therefore preferable.

An application was suggested where segments might be used - namely where one may wish to magnify part of a map and show greater details. A good meteorological example of this is observational plotting - although this is not a very commonly used application at ECMWF - whereby one has a hierarchy, and as one zooms in so more levels of the hierarchy are plotted. However, in order to use segments for this, all observations would have to be predefined in their relevant segments and this may well represent a considerable load on mass storage. A better approach would be to dynamically define the graphical information for observations as they are needed. This approach has the further advantage of being more flexible since there is no quantization of observations into levels of segments.

There was a discussion on the history of the GKS 'message' function. The outcome of the discussion was that we can use 'message' as our escape function for storing and using background files. Thus there is no need to upgrade beyond GKS level 1 and our objections to some of the concepts related to segments are no longer relevant.

3. Electrostatic plotter workstation: vector-to-raster conversion; techniques and relationships with GKS

There are many different vector-to-raster conversion algorithms (reference 7 discusses 11 methods) which can broadly be distinguished in two categories.

i) immediate rasterization

ii) storage of vectors in an intermediate vector file and delayed rasterization.

With immediate rasterization the complete raster image must be retained until all plotting is complete: this implies storage on a direct access device (disk) and updating portions of the complete raster image by swapping these portions between memory and the disks.

Normally the image is broken down into 'stripes' running across the entire width of the plotter paper. Wide stripes imply less I/O but greater field lengths, narrow stripes imply smaller field lengths but greater I/O.

An interesting technique is used at the Free University of Berlin (4) whereby the raster image is broken down not into 'stripes' but 'squares' each of 60 bits x 64 words. Since 64 Cyber words is one disk PRU, I/O can be optimised.

The Centre's experience on these various methods is that for our typical pictures - utilising a lot of short vectors for coastlines and contours - the Berlin technique is not beneficial. This is because although we use less CP time we use more I/O time. With the Centre's particular Accounting algorithm the extra I/O time adds more to the cost of the job than that saved by the reduced CP time.

However, it is not clear whether the Accounting algorithm is (or even should be) a valid measure for judging the efficiency of the vector-to-raster technique in terms of total machine utilisation. This is a particularly important question at the Centre where a high percentage of jobs are performing graphics.

Delayed rasterization has the advantage that it can even be performed in a separate job-step and then use the maximum available memory.

The Centre's Varian Basic Software uses an intermediate vector file with each vector broken into short incremental vectors. Intuitively, one would imagine this to be inefficient. This may well be so in general but for our applications with many short vectors it has proved to be an efficient solution.

Furthermore, it must be mentioned that because of the heavy use of selective erasure by the Contour Package (e.g. for Contour labels) we cannot pre-sort the vectors prior to rasterization. Consequently not all rasterization algorithms are applicable in our case.

Another point in favour of our current intermediate vector file approach is that it is the optimum as far as compression techniques are concerned. In other words, for our type of pictures, vector information takes less space than storing raster information, and incremental short vectors take less room than storing the start and end points of each vector.

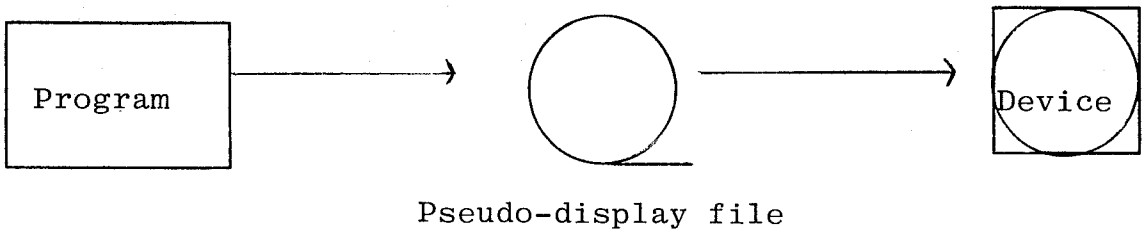
The above points refer to the Cyber - immediate rasterization may well be the optimum solution for the Cray. Furthermore, if our Accounting algorithm were to change, or if higher speed disks resulted in less of an overhead for I/O (as measured by the Accounting algorithm), the situation could change on the Cyber. Another factor is that for some applications one method may be more efficient, but other applications may favour the alternative method. Because of these factors we decided to implement two Electrostatic Plotter Workstations and let the user decide which to use.

4. Device independence and the Metafile concept

"A Metafile is a long term storage mechanism employing a sequential file. It must offer portability between machines and between applications and is essentially a 'slow' mechanism". (9)

Unfortunately, a lot of people confuse this concept of a metafile with what might be called a 'pseudo display file' i.e. a mandatory file between the application program and the device drivers. Such a 'pseudo display file' does embody part of the function of a Metafile as defined above, but is primarily the mechanism whereby graphics packages using 'pseudo display files' achieved device independence.

By contract device independence in GKS is not achieved through the use of files.



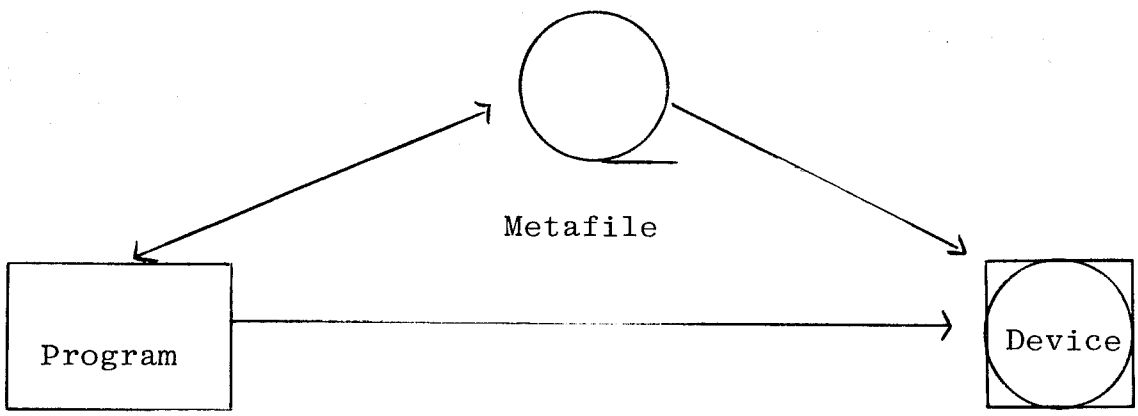


Figure 1

The difference between a mandatory pseudo display file and an optional Metafile. For clarity the device drivers and associated programs have been omitted.

may therefore be unacceptable to all users. Furthermore the limited pictures would have to be produced in addition to the full pictures.

A possible solution to this is via GKS segments or some similar construct with a hierarchical structure whereby some graphical information is marked as for 'high resolution' devices (e.g. Versatec plotters) and is therefore not viewed on the Tektronix. However, this is really a problem at the application level and is not a GKS problem.

Another way of looking at the problem is to consider the 3 categories of users:-

i) The user who produces many maps but does not know in advance which of all the pictures he needs to look at. Such users could perhaps use some other medium such as microfiche (which is especially cost-effective if multiple copies are made).

ii) Users with experimental research projects who don't know in advance what is required and who might use a pre-view technique to assess the experiments.

iii) Users who gain nothing from a pre-view step because they always require all pictures. As noted above 50% of output may be in this category.

As a longer term objective for saving paper, what is required is an interactive data base system with the ability to select pictures, zoom in on details etc.

The graphical meteorological workstation (AYDIN) is a step in this direction, but is a long term solution. Certainly something with local intelligence (such as the Aydin) is a prerequisite here. Such a 'black box' could be used for GKS distributed processing, but should be considered in conjunction with a fast link for good database I/O. For a truly interactive system one should consider adapting not to GKS but to the data.

Not only is the pseudo display file inefficient, but - as was pointed out by Mr. Buhtz (see Section 2.3) the adaptation to new hardware is not easy. The GKS concept of workstations for each specific device makes the adaptation to new hardware easier.

Two obvious uses of the Metafile at ECMWF were discussed.

i) As a way of transferring graphical data between the Cray and the Cyber. Although a Versatec workstation will be implemented on the Cray, no other workstations are envisaged, except a Metafile output workstation. This would allow Cray jobs to produce graphical output on any device supported by the Centre e.g. the FR80.

ii) As a way of producing graphical output on two different devices without the need to rerun the application program. For example, graphical output intended for the FR80 could first be viewed on the Tektronix - thus the Metafile removes the need for an FR80 simulator as currently employed.

5. Operational use of a Metafile

Discussion took place on the use of a pre-viewing facility at ECMWF primarily as a means of reducing our plotter paper budget (The Centre currently produces about 2km of plotter paper a week). Since vector-to raster conversion consumes about 45% of the total graphical activity, a pre-view facility was also seen as a means of saving CP resources.

However, if such a pre-view capability were forced on users we may well end up with using more CP resources. This is because 50% of plotter output comes from operational usage, and consequently all of this output is likely to be required. Pre-viewing such output on the Tektronix would be counter-productive both in terms of user's time but also CP resources.

Furthermore, in some cases a limit is already set on the amount of output produced - determined by the impossibility of users viewing more output. In other words, there may be more graphical activity if a way could be found to cope with the extra output on the human level.

It is highly unlikely that the Tektronic 4014's could cope with the present level of output in sufficient time - bearing in mind also that one can view plotter paper even when the Cyber is not available. The 4014's could be adapted to a higher throughput - but the only possible methods require manpower to make the changes. Two methods are:

i) Instead of using the PLOT10 software and 9600 baud lines, one could go straight to the internal 4014 bus and use speeds in excess of 92 Kbaud (8);

ii) A limited picture could be previewed on the 4014's with lower resolution, less contours etc. Note that these pictures will not be the same as on the plotter and

6. Adaptation of existing packages to GKS

The discussion centred on the adaptation of the Contour Package to GKS. A suggestion was made to keep the Contour Package exactly as it stands and use GKS instead of VARLIB, TEKLIB etc. Furthermore, a 'metafile' could then be used in all cases and this 'metafile' would need to be systematicallv pre-viewed by all users before deciding if they required paper output or not.

There were major objections to this approach which can be summarised as:-

i) Introducing GKS in this way gives users little knowledge that they are in fact using GKS and therefore they will not be attracted to it in place of the Varian Basic Software.

ii) Extending the graphics software this way could in fact lead to incompatibilities.

iii) The concept of pre-viewing is irrelevant to the adaptation of the Contour Package, furthermore it is not the best approach to take from the point of view of data analysis. (See the previous chapter for a discussion of the pre-viewing problem).

Another approach is to freeze the existing software and introduce a new Contour Package interfaced to GKS. All efforts would be made to maintain the major features of the present Contour Package in the new version but in a way which is fully compatible with GKS. In particular, heavily used routines such as CYLIND, CONTOUR etc. would retain the same name and calling sequence as at present.

There would be a long overlap period between the new and old software and thus users with old software would be catered for. As users develop new programs they would be guided to the new software and (hopefully)would be enticed by the new features which would be introduced.

Users cannot be made unaware of the use of GKS instead of the existing software, even if the Contour Package were to remain the same. For one thing the program size would alter and for another the names of routines would have to change (because GKS has a different set of functions compared to the Varian Basic Software). The change of names is important because users have their own sets of directives for the segment loader based on our 'model' set of directives. These directives name the routines explicitly and therefore would need to be altered - furthermore the structure of the segmented overlay might also have to be altered.

A major objection to the first suggestion concerns the use of the Contour Package routine NEWPIC which could not work in the same way with GKS as it does with VARLIB. This should not be seen as an objection to GKS but as a reflection of the fact that the Varian Basic Software is device specific.

GKS has the concept of three coordinate systems - the so-called World Coordinates (WC), Normalised Device Coordinates (NDC) and - for each device - Device Coordinates (DC).

Within the Contour Package the routine NEWPIC enables the user to place a new picture to the right of the old one, or above the old one if there is room. In other words, the Contour Package is here concerned with Device Coordinates.

Since one does not know a priori the size of the second picture, it is very difficult to fit this into GKS since we cannot use the Normalised Device Coordinates (NDC).

The solution is seen as a mixture of two approaches. Firstly, for page layout purposes we could go via NDC provided we know beforehand the size of all pictures on the page. Secondly, for paper saving (by putting pictures above previous pictures if there is room), one could make this a function of the device driver and by means of an escape function allow the user some control if he wants.

Yet another problem is that if the Contour Package were to maintain its existing interface to the low level software, but were to work with GKS instead of VARLIB,TEKLIB etc., then extra software is required which needs to be maintained.

The experience of the Graphics Section with users problems concerned with the routines USERON,USEROFF,CONTON,CONTOFF has led us to believe that to simulate the effect of these routines would be difficult. They were provided to allow users to add graphical information to images produced by the Contour Package. Instead of putting a lot of effort into simulating these routines, we feel it would be better to remove them from the Contour Package and use the methodology of GKS directly.

REFERENCES

(1) A. Lemaire, H. Watkins, A. Ducrot, E. Saltel
ECMWF Operations Department Technical Memorandum 27
"An evaluation of the GKS proposal for standards in Graphics with respect to the GSPC Core Proposal".

(2) A. Ducrot, A. Lemaire, H. Watkins
"A GKS Implementation for Meteorological Applications"
paper to be presented at Eurographics '81,Darmstadt.

(3) Workshop on ECMWF future Graphical System 10-13 July 1979
ECMWF, December 1979.

(4) J. Bechlars, "Speed and Memory Optimized Vector and Raster Graphics Generation", ECODU -29, 15-17 April 1980, Free University of Berlin.

(5) DIN 00 66 252: Information Processing: Graphical Kernel System (GKS) Version 5.2.

(6) ISO TC97/SC5/WG2
X3H3/80-55
Draft International Standard ISO/DIS+++
Information Processing:Graphical Kernel System (GKS)
Version 6.6

(7) W.R. Franklin "Evaluation of Algorithms to Display Vector Plots on Raster Devices"
Computer Graphics and Impage Processing 11,pp 377-397 (1979)

(8) G.A. Hamlin
"A High Speed Interface to Tektronix 401x Series Terminals". Computer Graphics, 14, No.4, p127 (March 1981)

(9) Minutes of the second meeting of ISO/TC97/SC5/WG2 - Graphics, Budapest, 21-22 October 1979