ALEPH 87-2
COMPRO 87-1

U.Berthon for
Graphics Working Group
15/12/86

PROGRESS REPORT OF THE GRAPHICS WORKING GROUP

1) PURPOSE OF THIS REPORT

The aim of this paper is to give an account of the work that has been done within the last months inside the graphics working group (GWG) and to show the direction in which we are going. In fact the end of the year 86 coincides with a certain milestone in our work. Up til now there is no clear evidence of our work for outside people, so we thought it might be good to take this opportunity to explain what we have done. This is particularly important since a good part of it goes beyond the range of graphics and should therefore be agreed upon also from outside the graphics group.

2) INITIAL SITUATION

The situation at the beginning of the year was the following:

hardware:

Due to the existence of equipment and various constraints, it was considered impossible to standardize on one hardware line (which would have been by far the most convenient and effective solution). The Aleph Graphics Guidelines of November 85 had pointed out various possibilities of graphics equipment, from simple terminals through to workstations up to sophisticated 3D-devices. Since no hardware standardization can be expected, a different way of ensuring the transportability of the application program had to be found.

application software:

There were (and still are) 2 independent and very different pieces of graphics software in Aleph

- the 2-dimensional plots developed by Hans Drevermann, working directly on the raw data and using colour and ingenious transformations for disentangling complicated structures [1] . This program was developed on a special image processing device, but can be used on all low-end graphics terminals
- the purely 3-dimensional program developed by the Wisconsin group and working under PIONS (thus running only on APOLLOS and MEGATEKS)[2]

underlying graphics package:

The decision to use a standard underlying graphics package and the choice of this package was the first and basic problem the GWG had to solve. Such a choice would give the possibility to transport the functionality of the two existing pieces of software in a same environment and practically on all suitable devices. It has to be noticed that the environment the 2 existing programs are working in is a purely graphical one, the software we are developing is part of a larger environment (see below).

3) REQUIREMENT GROUP

A group of physicists was appointed to formulate their requirements on graphics for Aleph in May 86 [3]. The main results of these discussions were the following:

a) Because of the high price, only a limited number of sophisticated 3D-devices will be available (20 to 30) . To ensure development and analysis facilities for many physicists, simpler devices will be used as well and are considered a possible way of doing graphics. Our data being essentially 3D-data, the "natural" way of seeing them must be available as well as elaborated 2D-plots chosen for special purposes. The aim should thus be to have a graphics program supporting 2D-terminals as well as 3D-ones, and allowing 2D-plots in a 3D-environment.

b) The graphics should be embedded in an interactive analysis environment. That means mainly two things:
- that one should be able to activate processes of the reconstruction program or user processes from inside this environment
- that the interactive user should be able to address in a very straightforward and flexible way any object in the data structure of the reconstruction program

c) Great emphasis was set on simplicity for the user. A flexible way of displaying things, for example with the use of windowing, was considered highly desirable.

4) CHOICE OF THE GRAPHICS PACKAGE

Taking into account
- the need for transportability
- the integration of 2-dimensional and 3-dimensional parts in one program

GKS-3D was virtually the only possible candidate. This choice was proposed by the GWG and has become official during the Aleph week in Munich. Details of the reasons that led to this decision, as well as the implications, are explained in [4].

5) FURTHER WORK:

5.1) GKS 3D

In fact, GKS-3D has not yet the status of an international (ISO) standard and only one implementation exists on the market (GTS-GRAL). This implementation has been tested in CERN and in Ecole Polytechnique, and work on it is continuing (see [5]). A certain amount of work was necessary to study the practical and financial background for the purchase of this package and verify that all necessary guarantees for the Aleph collaboration were taken as for purchase, distribution and disponibility on different devices. Since then, CERN has signed a contract with GRAL allowing the distribution of this software to outside institutes for a very low price. Remains the problem of non-existence of a 3D-driver for Megatek: this is about to be resolved by the fact that CERN is willing to contribute financially and/or by manpower in the writing by GRAL of this driver.

5.2) hardware

Since the Aleph Graphics Guidelines came out in November 85, the basis for any hardware choice has drastically changed, due to the decision on GKS and to hardware developments. These guidelines had therefore to be revised, taking into account the new facts (stating for example the availability of a GKS-driver). A new overview of possible hardware choices, based where possible on benchmarks, has been prepared by F. Etienne [6].

5.3) existing software

Further developments have been done on the PIONS-program by the Wisconsin-group. We think it is worthwhile to study its functionality in detail and then transport it to our environment (GKS and our data-structure and user interface).

Apart from improvements of the algorithms and methods, the 2D-plots of H.Drevermann have been transformed using a minimal part of GKS. They are gradually adapted to use the data structure of the reconstruction program. A GKS-version is presently included in the skeleton of the reconstruction program. The same remark as above applies here concerning transportation into our environment.

One point worth mentioning in this context is the following: GKS brings with it a certain number of fundamental concepts, and some of them are closely related to local intelligence given in some devices. They can thus be translated into hardware by the use of an appropriate driver. But for that the program has to be written adequately, this means that even for a program working correctly, a certain amount of work might be necessary for optimised usage of many devices.

5.4) the future graphics software

The greatest part of the work of the last months was the elaboration of the bases of this software. The questions that had to be answered were the following:

- how can the graphics routines access the data they are going to display?
- how can a user formulate his commands, i.e. how does he say which objects he wants to see on the screen and in which way they are to be displayed?

Since Aleph is using a data structure modelled according to the entity-relationship model, it seemed obvious to us that this had to be the basis of answers to these questions.

The way to answer the first question was to add graphics objects into the data structure used for the reconstruction program. A proposition about the integration of the graphics objects into the Entity-Relationship model used by Aleph was worked out, and was agreed upon by the graphics group in principle (see [7]). It foresees different ways of representing the objects (a simple sign or more complex drawings which might depend on some significant parameter), different ways of viewing them (3D or 2D with different type of coordinates, e.g. rho,z), of manipulating them on the screen (zoom,...) and changing their appearance(colour,...).

To the second question: To give the greatest possible flexibility to a user, a simple language must allow him to act interactively on any object in the data structure. This language has to be tailored to the data structure we are using. A proposition about this is presently under study. It should allow to specify in a flexible way, and in a formulation close to the English language, the next action to be done, modalities of this action and the objects one wants to act upon. Example: DRAW/RED all tracks with momentum>4. All sorts of abbreviations, as long as not ambiguous, will be permitted. Possibilities of compound statements (macros) will be provided as well as recording of actions for replay.

Example of a compound command :"DRAW" might include the following actions:

- reserve a window for display
- create the GKS-segments
- show them on the screen
- give the possibility of moving them

The user interface will as well offer menus as a convenient way of learning. In what menus are concerned, and more generally window-manager facilities, we are looking at the possibility of using already existing software.

The discussion about the internal structure of the program is not yet completely finished. The main ideas that seem to come out are the following [7] : the program will consist of a very thin layer of steering and a great number of independent modules. These modules are structured dynamically using macros [8]. They can be standard service modules, single reconstruction modules or user and analysis modules. Examples for service modules: display of an object, graphical manipulation of such a display (zoom, rotation etc) and others. There will be a general dialogue handler, catering for the command-line or menu-input. This dialogue handler can be called from any of the modules to get input.

6) PRESENT STATUS AND WORK IN PROGRESS

The investigations and discussions about the main issues are now nearly finished.

Our next step during the coming weeks will be to test the concepts described above by writing a small prototype according to them. The following stage will be to develop a skeleton that can serve as the framework of development of graphics and gradually grow into the final program. Some prerequisites for this are needed:

- complete and exact description of at least a coherent part of the reconstruction data structure
- complete and exact description of the graphics entities in this structure
- a working prototype of a language giving interactive access to objects in this data structure.

CONCLUSION

The last months were nearly exclusively dedicated to investigation and propositions . This was a necessary phase of our work. Elaboration of the main concepts is now finished. Written propositions have come out or will come out about

- modelisation of graphical objects in the reconstruction data structure [7]
- an interactive language of access to objects in the data structure
- the overall structure of the environment in which the graphics routines are supposed to work [7]

A prototype shall test and illustrate these concepts in the near future. Once we will have proved that the framework is working in its principles, we shall develop a skeleton which will allow a gradual development of the facilities and at the same time guarantee a continuous usage.

Two things have to be stressed:

- the difficulties related to the geographical dispersion of the active members of the group
- the fact that the subject we covered is much larger than only the graphics problem and will have to be taken in charge correctly and outside of the graphics group

If we want to be able to come to a good realisation rapidly, we need more full-time manpower in the graphics group.

The work that has to be done now is the following:

- describe the graphics parts of the data structure completely
- define, in conjunction with the physicists concerned, what are the representations to be used for each subdetector
- choose the menu and window manager package to be used
- continue the writing of the utility modules like dialogue handler, command language interpreter, viewing server etc
- write the basic graphics modules

REFERENCES:

[1] Hans Drevermann
CFT: a program for 2-dimensional display of ALEPH events , May 86

[2] A. Caldwell,M.Hildebrandt,J.Hilgart,M.Mermikides
A 3-D Interactive Display Program for ALEPH, Users' Guide, May-June 1986
and:
Michael Hildebrandt:
Technical Guide for ALEPH display program 2/5/86

[3] Graphics Requirements Group (Bowdery, Brandt, Cordier, Drevermann, Fernandez, Knobloch, Mermikides, Rolandi, Schlatter, Videau)
Requirements for a Graphics and Interactive Programming Environment
ALEPH 87-1, NOTE 1

[4] U.Berthon
memorandum about decision on graphics standard, 10/09/86

[5] A.Finch
GKS-3D - Coordinates, Windows, Viewports and Viewing, 17/06/86

[6] F.Etienne
Graphics Devices for ALEPH, paper in preparation

[7] U.Berthon,S.Fisher,H.Videau
A Proposal for an Interactive Analysis Environment including Graphics
ALEPH 87-3, COMPRO 2

[8] C.Bowdery
SYSTEM MACROS: A New Concept in Graphics Coordination, 17/10/86
Aleph 86-117