SOFTWARE—PRACTICE AND EXPERIENCE, VOL. 14 (3), 277–290 (MARCH 1984)

An Interactive Data Visualization System*

DANIEL THALMANN

Département d'informatique et de recherche opérationnelle, Université de Montréal, Montréal, Canada

SUMMARY

GRAFANA is an interactive program, which allows the user to produce drawings based on data stored on a disk file. The major advantage and originality of the system is that the user can interactively determine the visual effect of a drawing and immediately see it on the screen. Moreover, any drawing produced by GRAFANA can still be edited by a general-purpose graphics editor. This system allows the user to obtain precisely the right drawing for any book, paper or report.

GRAFANA has been implemented in MIRA-2D, a graphical Pascal extension. It is available for different machines and kinds of graphical devices.

KEY WORDS Plots Graphics edition Data visualization Interaction Command-directed system Pascal

INTRODUCTION

Pictures are a primary information-carrier. In fact, the importance of graphics for conveying information arises from the nature of the human eye. As discussed by Myers,[1] graphical images may be processed in a single viewing, whereas text fails to use fundamental human abilities. A graphical plot has greater information density than a single table of values. Trends, comparisons and decisions can often be obtained by looking at a histogram or a curve.

For many years, scientists and business people have been able to obtain computer generated plots based on data files. The most common way of producing such plots is by the use of a specialized library which contains a subroutine to draw curves and another for histograms. These subprograms have been developed for a batch environment.[2, 3] They require a lot of parameters and if a user wishes to modify these parameters interactively and see the effect produced, he/she has to write his/her own interactive program. Moreover, these software packages are generally limited to Cartesian plots, histograms and possibly pie charts.

Another limitation of this software is that it is not possible to edit a plot, once it has been completely produced.

This paper presents an interactive system which allows the user to produce Cartesian plots, histograms, network diagrams, pie charts, 'size' charts and 'number' charts. Although the diagrams represent data stored on text files, interactive commands allow the user to prepare attractive diagrams by making a decision based on the visual effect. The user may modify a diagram until it is considered acceptable. Moreover, such a diagram can still be edited using a general-purpose graphics editor.

* This work was supported by the Natural Sciences and Engineering Research Council of Canada.

0038–0644/84/030277–14$01.40 *Received 15 October 1981*

THE GRAFANA SYSTEM

The GRAFANA system is an interactive program, which allows the user to produce drawings based on data stored on disk files. These drawings are illustrations of the data and there are six kinds of drawings which may be represented:

(i) Cartesian plots
(ii) histograms
(iii) network diagrams
(iv) pie charts
(v) 'size' charts
(vi) 'number' charts.

The system offers a natural, efficient and friendly dialogue with the user. The goal of this dialogue is to produce graphical output which is convenient to the user. This means that the user can interactively modify a small number of parameters which change the visual effect of the drawings.

GRAFANA is a command-directed system. The user selects commands from a set of instructions and types them on the keyboard of a graphics CRT. These commands are displayed in an alphanumeric area at the bottom of the screen. The current drawing is displayed on the rest of the screen.

The system allows the user to dynamically decide which is the current data file. A specific command (OPEN) interactively opens a file and makes it current. All subsequent commands apply to a drawing based on the current file. At any time, it is possible to save the current drawing on a disk file. As the data do not take the same form for each type of drawing, three kinds of files are available and a few commands are specific to a type of drawing. These commands and the format files are presented in the next sections.

HISTOGRAMS AND CARTESIAN PLOTS

Histograms and Cartesian plots (which we call 'curves') use the same kind of data file. This is a text file with the following format:

CH
number of variables n
$\langle v_{11} \quad v_{12} \cdots v_{1n} \rangle$
$\langle v_{21} \quad v_{22} \cdots v_{2n} \rangle$
$\langle v_{m1} \quad v_{m2} \cdots v_{mn} \rangle$

The user may give a name to each variable and specify a subrange for this variable. Only values in the range will be considered in histogram or curve construction, e.g.

NAME 1 *year*
SUBRANGE 1 1970 1980
NAME 2 *inflation_rate*
SUBRANGE 2 0 25

The first parameter of each command is the index i of the variable v_i.

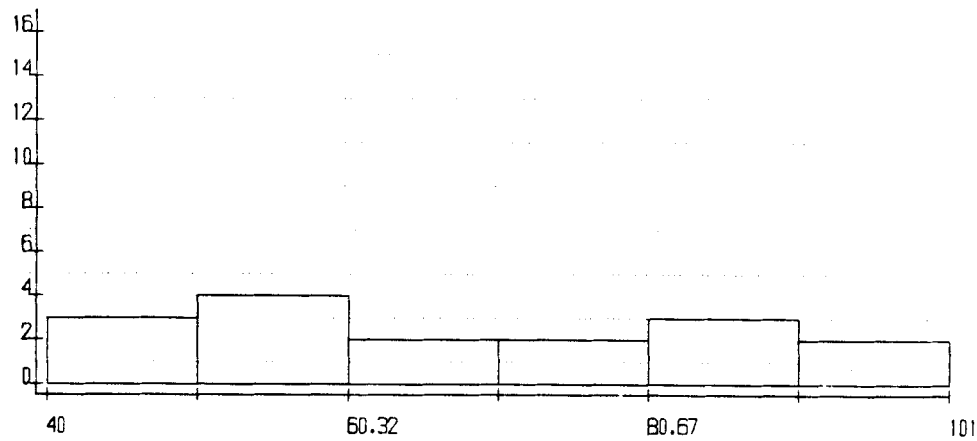
Histograms may be interactively constructed for any variable and the number of channels has to be given, e.g.

HISTO *inflation_rate*
CHANNELS 5

Figure 1 shows an example of a histogram with the data file and the user interactive commands.

Curves may be generated for any pair of variables; selection of graduations on both axes is made through the command AXES *nx ny*. Smooth curve fitting is obtained using the Akima[4] method of interpolation, e.g.

```
CURVE inflation_rate year
AXES 5 5
INTERPOLATE 3
```

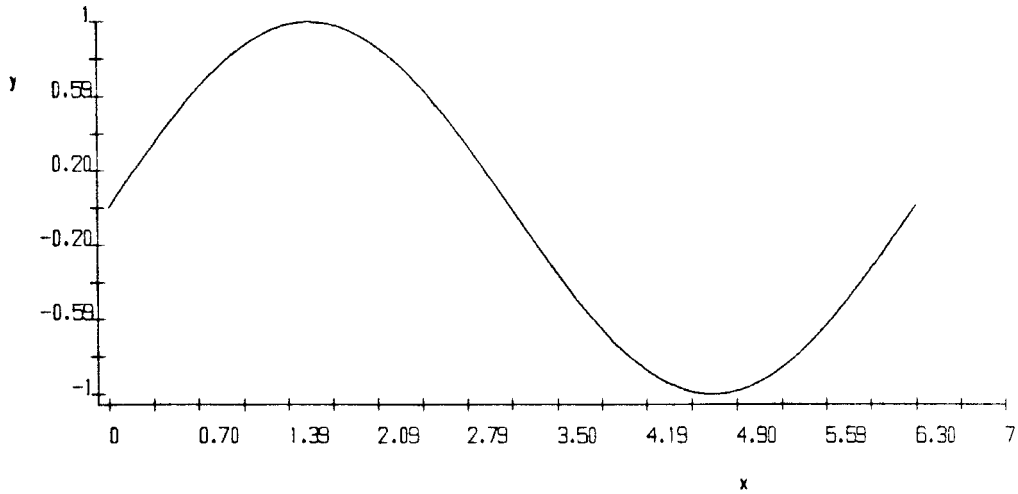



```
OPEN H 1 GRADES
SUBRANGE 1 40 100
NAME 1 GRADES
CHANNELS 6
HISTO 1
DRAW
```

```
CH
1
45.6
56.7
89
56.7
78.9
87.5
56.4
60.8
76.5
45.6
43.2
89.3
59.5
60.9
92.1
95.8
```

Figure 1. A histogram with the corresponding commands and data file

Figure 2 shows an example of a curve with the corresponding commands. The data file is displayed in Figure 3 with the Pascal program that produced it. This is the only way of representing an analytical function, because there is no feature in GRAFANA for producing a curve from an algebric expression.




```
OPEN C 2 DATA
SUBRANGE 1 0 7
SUBRANGE 2 -1 1
NAME 1 X
NAME 2 Y
CURVE Y X
AXES 20 10
INTERPOLATE 5
DRAW
```

Figure 2. An example of curve with the corresponding commands

```
PROGRAM SINE(OUTPUT,DATA);
CONST
   PI=3.1416;
   STEP=0.3;
VAR
   DATA:TEXT;
   X,Y:REAL;
   I:INTEGER;
BEGIN
   REWRITE(DATA);
   WRITELN(DATA,' CH');
   WRITELN(DATA,2);
   X:=-0.3;
   FOR I:=1 TO 22 DO
     BEGIN
       X:=X+STEP;
       Y:=SIN(X);
       WRITELN(DATA,X:10:4,Y:10:4)
     END
END.
```

```
CH
         2
    0.0000    0.0000
    0.3000    0.2955
    0.6000    0.5646
    0.9000    0.7833
    1.2000    0.9320
    1.5000    0.9975
    1.8000    0.9738
    2.1000    0.8632
    2.4000    0.6755
    2.7000    0.4274
    3.0000    0.1411
    3.3000   -0.1577
    3.6000   -0.4425
    3.9000   -0.6878
    4.2000   -0.8716
    4.5000   -0.9775
    4.8000   -0.9962
    5.1000   -0.9258
    5.4000   -0.7728
    5.7000   -0.5507
    6.0000   -0.2794
    6.3000    0.0168
```

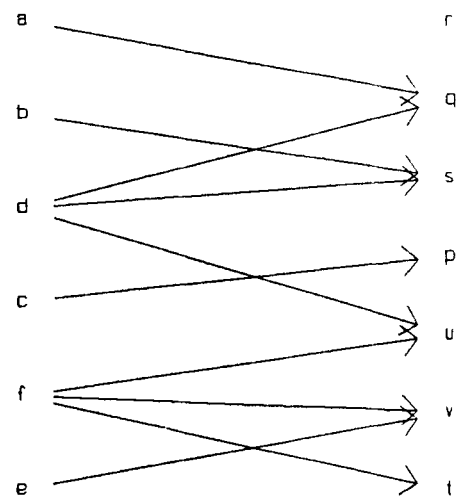
Figure 3. The data file and the Pascal program corresponding to Figure 2

NETWORK DIAGRAMS

Network diagrams (or graphs) can have one or two dimensions; they are stored on data files using a matrix representation:

if $\langle x_i, y_i \rangle \in$ R **then** $m_{ij} = 1$ **else** $m_{ij} = 0$

The matrix is preceded by G1 (for unidimensional graphs) or G2 (for two-dimensional graphs) and the names of the nodes. Figure 4 shows an example of a two-dimensional graph and the corresponding data file.




```
G2
A,B,C,D,E,F.
P,Q,R,S,T,U,V.
0100000
0001000
1000000
0101010
0000001
0000111
```

Figure 4. An example of two-dimensional graph and the corresponding data file

It is very difficult to make a good drawing of a network. The problem is to avoid collisions between arcs, nodes and arrows, which can make the network completely unreadable. This problem can be solved for specific graphs such as trees,[5] but a very general solution which allows the user to produce aesthetically pleasing graphs automatically is quite unrealistic. However, in an interactive environment, the user does not need an automatic tool; he/she may build a graph by trial and error because the results are immediately visualized.

GRAFANA provides the user with a command

DISPOSE $\begin{Bmatrix} C \\ L \\ R \end{Bmatrix}$

which determines the positions of the nodes in the graph.

The C option produces a circular arrangement; the L option gives a linear arrangement and the R option permits a random arrangement. Figure 5 shows three different outputs for the same data file. Once the user has chosen the kind of

arrangement required, two commands are available to modify the visual impact of the graph.

The command ORDER *n* applies a circular permutation of *n* positions on the nodes. Figure 6 shows the effect of ORDER 3 on the graph of Figure 5(c).

Often, only one node has to be moved to produce an attractive looking graph. This has been made possible by the command MOVE which interactively enters graphical data using the terminal cursor. Figure 7 shows a graph before and after the use of the command MOVE.

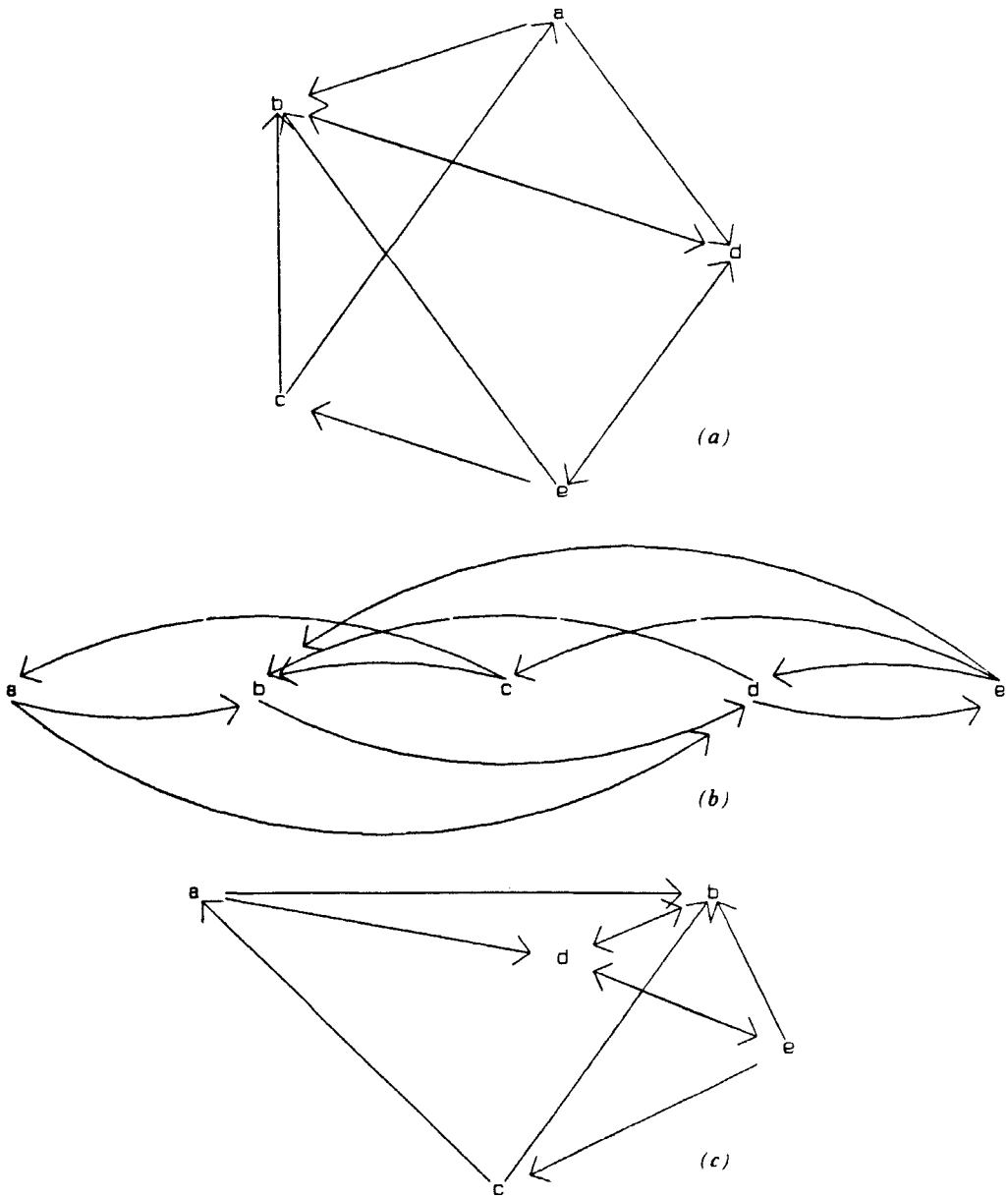


Figure 5. (a) A circular arrangement. (b) A linear arrangement. (c) A random arrangement

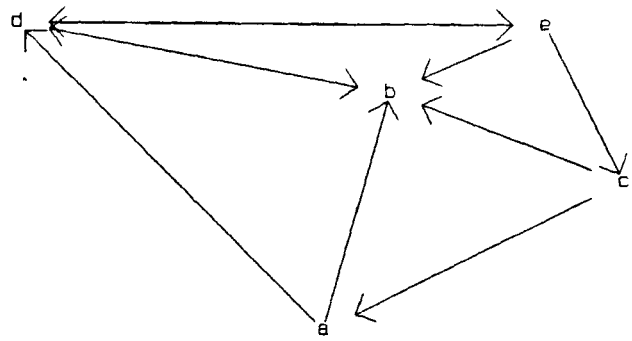


Figure 6. The effect of ORDER *3 on the graph of Figure 5(c)*

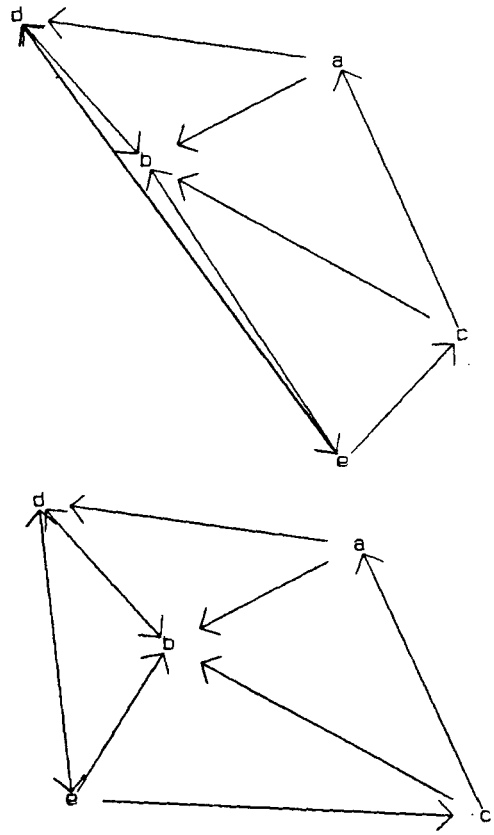


Figure 7. A graph before and after the use of the command MOVE E

MEASURE FILES

Much of the information currently presented in books or newspapers has the same basic structure, consisting of a list of names with an associated value. We mention the following examples:

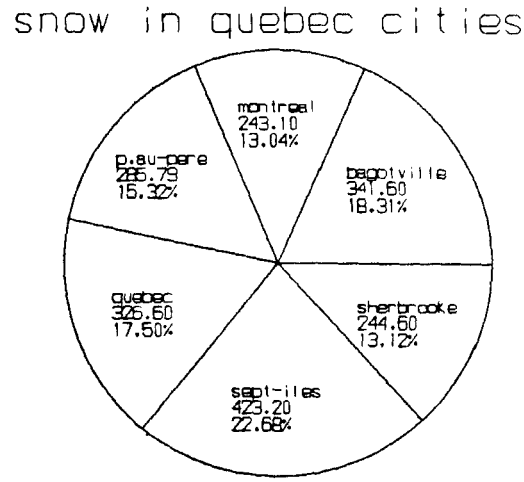
(a) the oil production of the ten major producers

(b) the world's six largest companies by profit
(c) the seven safest airline companies (in terms of number of passengers/number of accidents).

This kind of information may be represented in different graphical forms. GRAFANA allows the user to choose three different types of drawings based on the same data file. This latter is a text file with a letter M followed by the list of names and values.

The three types of drawings are as follows:
(1) pie charts
(2) 'size' charts
(3) 'number' charts.

The pie chart is the most well-known drawing: it is a circle, which is divided like a pie into slices, whose area is proportional to the values in the data file. Figure 8 shows an example of a pie chart with the corresponding data file.




```
M
SHERBROOKE 244.6
SEPT-ILES 423.2
QUEBEC 326.6
P.AU-PERE 285.79
MONTREAL 243.1
BAGOTVILLE 341.6
```

Figure 8. A pie chart with the corresponding data file

A 'size' chart is a chart where each value is represented by a picture, whose size is proportional to the value. This kind of chart is created using the command

$$\text{SIZE} \quad \begin{Bmatrix} 1 \\ 2 \end{Bmatrix} \quad \langle \textit{filename} \rangle$$

There are two options: with option 1, the height of the Figure varies with the values; with option 2, both height and width are dependent on the values. As may be seen in Figures 9 and 10, option 2 is more aesthetically pleasing. However, option 1 gives a more realistic view, because the area is proportional to the values.

The filename, which is present in the SIZE command, is the name of the file that contains the unit picture. This latter can be any two-dimensional picture created by

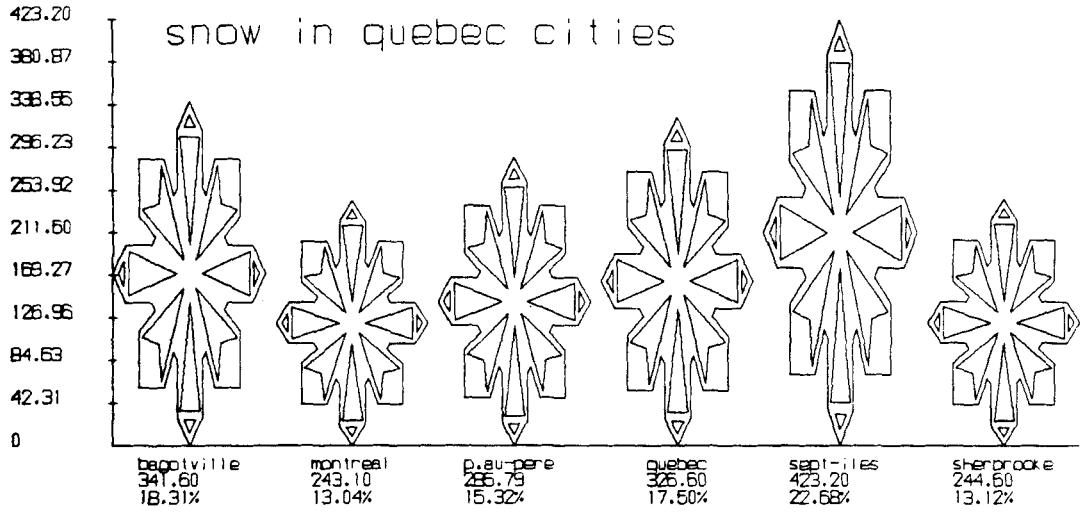


Figure 9. A 'size' chart with option 1 (the height varies)

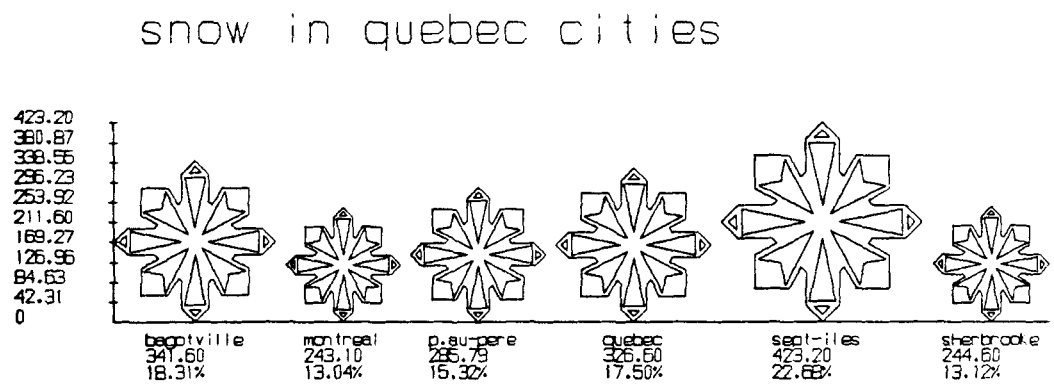


Figure 10. A 'size' chart with option 2 (both height and width vary)

the general-purpose graphics editor GRAFEDIT[6] or by a MIRA-2D[7] program. It is, of course, recommended that the picture bears some relation to the values to be represented.

A 'number' chart is a chart in which each value is represented by a small number of similar pictures. This number is directly proportional to the value that it represents. This kind of chart is created by the command NUMBER ⟨*weight*⟩⟨*filename*⟩ where the first parameter is the weight of the unit picture and the filename has the same meaning as for the 'size charts'. Figure 11 shows an example of a 'number' chart.

For the three kinds of drawings, the values are presented in order of appearance in the data file. This may lead to a very confusing drawing as in Figure 12. In this case, the OPTIMIZE command allows the user to obtain better results, as shown in Figure 13.

Figure 11. A 'number' chart

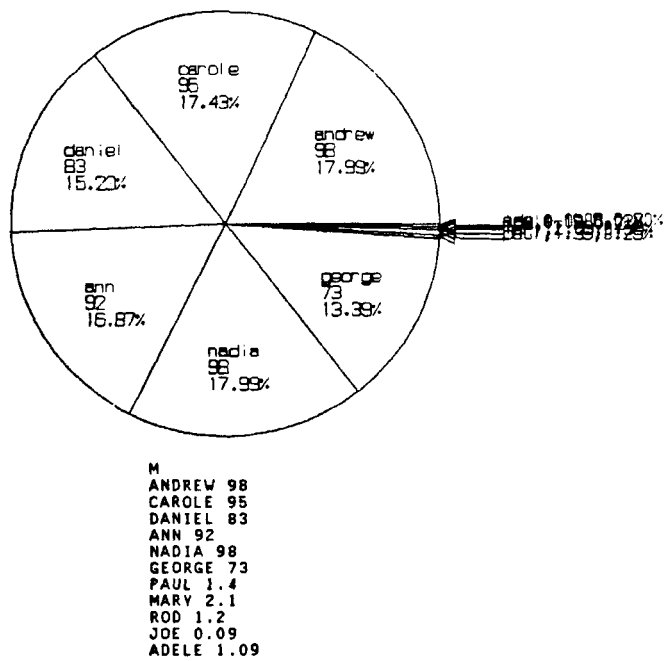


Figure 12. A very confusing drawing

GRAPHICS EDITING

GRAFANA allows the user to interactively add a title to his/her drawing and redefine the scale of the drawing.

However, frequently the user would like a modification of the final drawing. GRAFANA does not provide the user with such a facility. This is why we have

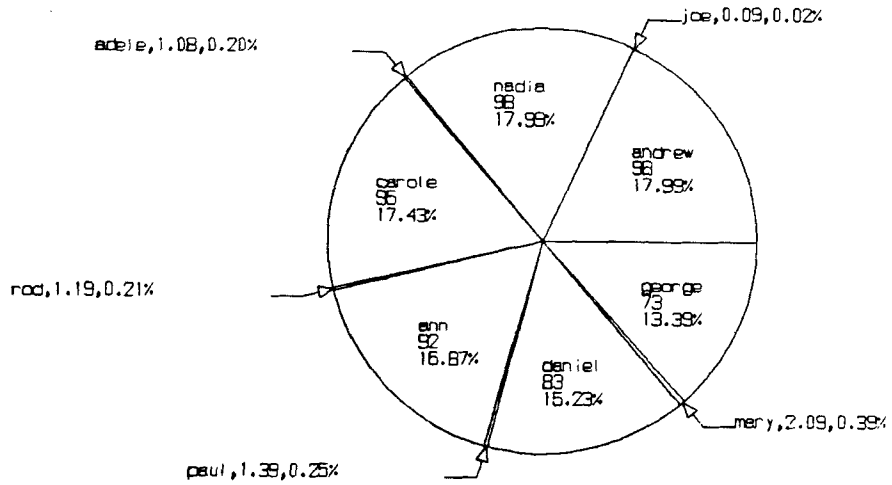


Figure 13. The effect of the OPTIMIZE *command on Figure 12*

introduced into GRAFANA a command which saves a drawing on a file in a form which is completely compatible with the general-purpose graphics editor GRAFEDIT. In an interactive editing session, the user may insert, change or erase, a line, a value, a picture or a title on a drawing produced by GRAFANA.

As the final goal of the user is the production of a document, any diagram which has been saved by GRAFANA and possibly edited by GRAFEDIT may be hardcopied.

IMPLEMENTATION

GRAFANA is a 2500 source lines program. It has been written in MIRA-2D[7], a graphical Pascal[8] extension based on abstract graphical data types.[9] The program consists of five main modules:

(i) a monitor which checks the system and executes the commands
(ii) a parser which analyses the commands and their parameters
(iii) a user interface which is responsible for the user-machine dialogue
(iv) a data structure handler
(v) an input/output module.

The program runs on any machine for which MIRA-2D is available. At present, there are versions on CDC Cyber and VAX 11, but as MIRA-2D is implemented using a Pascal preprocessor, the effort required to move it to a new environment is not unreasonable. The time required can be estimated to be one man-month, depending on the Pascal compiler existing on the target machine.

GRAFANA may be used with various kinds of terminals: HP2648A (black and white), DEC GIGI, AED 767, and TEKTRONIX 4027 (colour). Hard copies may be produced by VERSATEK, TRILOG and PRINTRONIX printers and HEWLETT-PACKARD and CALCOMP plotters.

CONCLUSION

GRAFANA has been designed to help people who need drawings to present their data. The system offers different kinds of drawings based on data stored on a *disk file*.

The major advantage and originality of the system is that it allows the user to interactively modify the visual effect of the drawings. This point is fundamental and users of the system find it much more useful than a traditional plot package. They can obtain precisely the right drawing for their papers, books or reports. They avoid spending too much time on the process of modifying parameters in calling sequences of subroutines. Moreover, as the final drawings produced by GRAFANA can be edited using a general-purpose graphics editor there is almost no limitation to the system.

ACKNOWLEDGEMENTS

The authors would like to express their gratitude to the students who worked on this project: M. Burgess, P. Lafortune, A. Larouche and G. Mainville, and also to Ann Laporte who revised the English text.

APPENDIX

General commands

OPEN $\left\{\begin{matrix} \text{C} & n \\ \text{G} & n \\ \text{H} & n \\ \text{M} & \end{matrix}\right\}$ ⟨*filename*⟩

opens a data file and set one of the four modes:
C : curves (*n* is the number of variables)
G : graphs (*n* is the dimension of the graph)
H : histograms (*n* is the number of variables)
M : measures

DRAW
draws the diagram produced by the preceding commands

END
terminates the session

PAGE
clears the screen

SAVE ⟨*filename*⟩
saves the current diagram in a file which is compatible with the graphics editor GRAFEDIT

TITLE ⟨*string*⟩
adds a title

VIEWPORT x_{inf} y_{inf} x_{sup} y_{sup}
defines a viewport in normalized co-ordinates

Commands for histograms and curves

SUBRANGE *n inf sup*
defines the subrange [*inf*...*sup*] for the *n*th variable

NAME *n* ⟨*identifier*⟩
gives a name to the *n*th variable

Only for histograms

HISTO $\left\{\begin{matrix} n \\ \langle identifier \rangle \end{matrix}\right\}$
selects the histogram for a given variable

CHANNELS n
selects the number n of channels

Only for curves

CURVE $\left\{\begin{matrix} n_1 \\ \langle identifier \rangle \end{matrix}\right\}\left\{\begin{matrix} n_2 \\ \langle identifier \rangle \end{matrix}\right\}$
selects one variable as a function of the other one

AXES nx ny
selects the number of graduations on both axes

INTERPOLATE n
applies the Akima interpolation method with n points between two existing points.

Commands for graphs

DISPOSE $\left\{\begin{matrix} C \\ L \\ R \end{matrix}\right\}$
selects the arrangement of the nodes
C : circular arrangement
L : linear arrangement
R : random arrangement

MOVE ⟨*identifier*⟩
moves a node to the place pointed to by the user

ORDER n
applies circular permutation of n nodes

Commands for measures

PIE
creates a pie chart

SIZE $\left\{\begin{matrix} 1 \\ 2 \end{matrix}\right\}$ ⟨*filename*⟩
produces a 'size' chart based on the picture stored in the file; the variation may be in one or two dimensions

NUMBER ⟨*unit*⟩ ⟨*filename*⟩
produces a 'number' chart based on the picture stored in the file; the value of one unit may be fixed

OPTIMIZE
arranges the sequence of values to obtain the best graphical results.

REFERENCES

1. W. Myers, 'Computer graphics: a two way street', *Computer*, **13** (7), 49–58 (1980).
2. J. Butland and S. D. Butland, 'An easy to use graph drawing package', *Computer*, **13** (2), 69–80 (1980).

3. R. A. Earnshaw 'Graph plotting in ALGOL 68-R', *Software—Practice and Experience*, **6**, 51–60 (1976).
4. H. Akima, 'A new method of interpolation and smooth curve fitting based on local procedures', *Journal of ACM*, **17** (4), 589–602 (1970).
5. J. G. Vaucher 'Pretty-printing of trees', *Software—Practice and Experience*, **10** 553–562 (1980).
6. N. Magnenat-Thalmann *et al.*, 'GRAFEDIT: a general-purpose interactive graphics editor'. *Computers and Graphics*, Pergamon Press, Oxford, 1981.
7. N. Magnenat-Thalmann and D. Thalmann, 'A graphical Pascal Extension Based on Graphical Types', *Software—Practice and Experience*, **11** 53–62 (1981).
8. K. Jensen and N. Wirth *PASCAL User Manual and Report*, Springer-Verlag, Heidelberg 1974.
9. D. Thalmann and N. Magnenat-Thalmann 'Design and implementation of abstract graphical data types', *Proc. COMPSAC '79*, Chicago, IEEE Press, 1979, pp. 519–524.