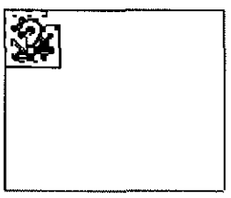
[Previous] [Next] [Contents]



METVIEW - Plot Module

Design Document

Version 1.3 - June, 1997

Purpose

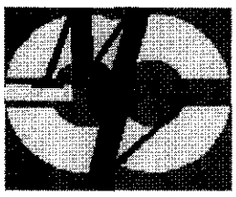
The purpose of this document is to describe a new module for visualisation and plotting meteorological data in Metview. The document contains a functional description (including the user interface), a description of the interfaces between this module and the other METVIEW modules, and a design strategy for the internal module structure.

Caveat

This is a working document. According to modern programming practices, software develpment should be done in an evolutionary way: the system should grow from an initial design and implementation through repeated redesign and implementation. Therefore, some parts of this document may be incomplete at any given moment, and the document will be reviewed as the implementation evolves.

Table of Contents

[Previous] [Next] [Contents]



METVIEW - Plot Module

1. Introduction

The development of a new visualisation and plotting module for METVIEW, which is upwards compatible with the current `VisMod`, is a logical consequence of a software project's normal development curve. The Software Engeneering literature contains many reports that document that an interactive program can only be completely specified when a working version is running.

Therefore, the standard recommendation for interactive program development (see, for example, Brooks [1992]) is that an interactive system should be re-written after a working version is submitted to substantial user trials, which provide an acessment of its usability and adequacy. The working version is therefore considered as a basis, in terms of functionality and interface, for a second version, which is then designed to account for performance, maintenance and reliability.

The current `VisMod` was developed, in its major parts, on the period 1992-1995, whilst the METVIEW design was evolving. During the development period, a number of additions were made to the software, which had not been foreseen in the original design. These changes inavoidably affected the software's structure and have made it necessary for a second version to be developed.

This document contains the design for this second version of the visualisation and plotting module in METVIEW, based on the following assumptions:

- The interface and functions should be preserved (and revised, when needed), based mostly on an usuability analysis of the current version.
- The interface to the `GenApp` module, based on requests and the drag-and-drop widgtes, should be maintained.
- The interface to the `MagProc` module should be revised.
- The internal structure should be completely reviewed. Altough reuse of X-Window related widgets will be attempted, a new design is needed.
- The resulting software should be portable across a wide range of UNIX-based operating systems (IRIX, DEC/OSF, HPUX, Solaris, AIX and Linux), should be easy to be maintained.
- Given user expectation of typical interactive software, efficiency considerations will play an important role in the PlotMod system design.

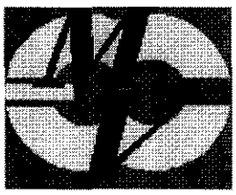
The new module will provisionally be called `PlotMod`. The idea of developing the module under a different name stems from Baudoin Raoult, and allows the current `VisMod` and the new `PlotMod` to be available concurrently, and thus make the transition a smoother one.

The document structure is as follows:

- Chapter 2 provides an user perspective of the functionality of the current `VisMod`, based mainly on the users perspective, but also on the experience of the Graphics Group at ECMWF and CPTEC.
- Chapter 3 presents a proposed object-oriented design strategy for the new module. This design has taken into account, whenever feasible and application, the new developments in the Software Engineering area, especially those related to the emerging discipline of *software patterns*.
- Chapter 4 contains a description of the relation of `PlotMod` to the othe METVIEW modules, with a view of maintaining compatibility with the current METVIEW implementation, especially in the processing of requests and macros, the relation to the `GenApp` module ("drag-and-drop"), and the relation between `PlotMod` and the `MagProc` module.

- Chapter 5 describes the software implementation componentes of `PlotMod`.
- Chapter 6 contains some concluding comments and acknowledgments.
- An Appendix contains the headers of the C++ classes used in the development of `PlotMod`.

[Previous] [Next] [Contents]



METVIEW - Plot Module

2. User's Perspective of Current Version

2.1 Introduction

A limited usability test was conducted at ECMWF to acess the current uses of METVIEW, especially in what relates to visualisation and plotting issues. Each user was asked to show how he uses METVIEW in a typical working session, explain what problems he is currently facing, indicate possible improvements and corrections he needs or requires.

On average, each interview lasted one hour. Inevitably, the users expressed opinions over the complete METVIEW software. For the purpose of this document, this section concentrates on the the topics involving the current VisMod.

The users interviewed were:

- Meteorological Operations: Anders Peerson, Antonio Garcia-Mendez, Brian Norris, Frederic Atger and Rogerio Bonifacio.
- Research Department, Physical Aspects Sections: Christian Jacob, Pedro Viterbo.
- Research Department, Data Assimilation Section: Erik Andersson.
- Research Department, Satellite Section: Elisabeth Gerard.

This sections also draws on the experience of CPTEC users, including:

- Meteorological Operations: Prakky Satyamurti, Eugenio Almeida, Paulo Etchichury.
- Research, Model Division: Jose Bonatti.

This section presents the users considerations, grouped by topics. In what follows, software similar to METVIEW, such as Vis5D, GRADS and McIdas, will be mentioned, whenever it is relevant to compare their functionality with that of METVIEW. These mentions are not intended to provide a basis for claims of superiority of one software over the other, but rather as a result of user experiences and opinions, especially when the user has had access to different systems.

2.2 General METVIEW Use

Overall, there seems to be some significant trends in METVIEW usage:

- A general satisfaction with the user interface, seen as "user-friendly" and having a small learning curve.
- The ever-growing use of the macro language as a substitute for FORTRAN programs for manipulation and post-processing of model data.
- An emphasis on shaded contour maps as the most common method of generating pictures from model data.
- A requeriment that METVIEW should provide as close to a WYSIWYG ("what you see is what you get") performance as possible, achieving a correspondence between screen views and plotter output.

- An overall concern about graphical performance, especially in terms of the time it takes to draw (or redraw) 2D maps.
- A requirement to improve the facilities for displaying and manipulating satellite imagery.
- A requirement to allow user more control of how applications display data, including the possibilty of displaying different perspectives from the same data set.

The ubiquitous topic of the "paper-less office" was mentioned on most discussions with users. As a general view, it was felt that the workstation graphical capabilities gave the user a chance of experimenting various different visual definition, before selecting those that were fit to print. There were users (especially on Meteorological Operations) in which case the quantiity of screen-generated output greatly exceeded the number of plots produced.

Therefore, a typical METVIEW user cycle could be expressed as:

- Preparation of a macro program
- Running a macro
- Visualization of the output
- Plot preview of some fields
- Plotting chosen maps.

Specific comments for each issue are presented below. In what follows, the main emphasis is placed on the issues related to the current VisMod, altough it is inevitable that some issues are related to other METVIEW modules.

2.3 User Requirements for a New VisMod

The usability test concentrated on questions regarding the users satisfaction and wishes as regards the `VisMod` module, which are outlined below.

2.3.1 Plot Window Layout and User Control

In general, it was felt that `VisMod` does not allow the user sufficient control over the output displayed. The interaction of `VisMod` with applications other than standard 2D contouring (and area fill) is not satisfactory; dropping application icons into a plot window may generate undesirable results. Therefore, currently most users use separate plot windows for displaying different application results. The unpredictabilty of `VisMod` behaviour is a very negative aspect and the new module will need to have a well-established behaviour pattern.

2.3.2 Graphics Performance

It has been acknowledged many times by all METVIEW and MAGICS users that the graphical quality of output is extremely good, and this has been a major positive aspect of METVIEW. However, there is concern from users on the graphics performance of METVIEW, especially as regards interactive screen plots.

Compared with similar packages (such as GRADS), METVIEW is one order of magnitude slower when drawing a shaded contour plot. In one experiment performed at CPTEC, with data already on the local disk, the same shaded contour took 3 seconds on GRADS and 15 seconds on METVIEW.

The issues involved here are not simple and a straightforward answer to this problem is not possible. One of the problems is that when producing screen output, interactive users prefer speed to quality,. The inverse rule applies to generation of paper output. As a general guideline, it was felt that the new `PlotMod` should allow for a compromise between screen output and paper output, allowing different graphics engines to be associated with it.

2.3.4 Animation

Animation in METVIEW is used not so much as a mean of sequence visualisation but as a way of rapidly browsing through a set of files. In this respect, there is an overlap between the functionality provided by the slider in normal operation mode for a plot window (where a user would browse all his graphics), and the animation is frame-by-frame mode.

Overall, it was felt that the animation could be merged with the visualisation, since it not felt to provide a significant benefit from being a separate function.

2.3.5 Imaging

The image manipulation functions available on the current `VisMod` fall short of the typical ones necessary for most users which work with images. Whilist a full-blown image processing module for METVIEW would require a significant developement, the gap between what is available and what most users would regard as sufficient is not so great, and includes:

- Provision for image manipulation functions in macros.
- Interactive interfaces for defining enhancement curves.
- Support for colour imaging in METVIEW, as colour composites and false-colour enhancement.
- Facilities to combine NWP fields (e.g., cloud cover) with satellite images.

2.3.6 Colour Manipulation and Presentation

Most users felt that the colour coding in METVIEW did not satisfy their requirements. METVIEW should, in their view, provide a continous-tone colour pallete from which a user could choose the desired set of colours (typically by providing a start and an end colour). The legend should also follow the same convention and be presented as a continuous-tone legend, rather than a set of colour boxes.

Typical systems which provide continuos colour choice and legend to the user are `Grads` and `Vis5D`. The fact that these systems do not always allow complete flexibility as METVIEW does (e.g., for defining the direction in which the colour wheel is traversed) is less important, in the user's perspective, than a wider choice of colours.

2.3.7 Compatilibity between batch and interactive processing

There are currently two different ways in which graphical output is produced in METVIEW currently, one by defining a `PlotWindow` and the other by defining a `SuperPage`. It is a strong wish from the users that these commands are unified and that the current differences in batch and interactive processing removed.

2.3.8 3D Visualisation

3D visualisation is coming of age in operational and research meteorology and many users are now starting to rely on it as a means of obtaing important information for large data sets. ECWMF has started to use 3D on a daily basis, especially from examining ensamble forecasts and propagation of forecast errors. The most widely used 3D visualisation packages on the meteolorological community is Vis5D. Given that it already provides a substantial set of functions and that it is freely available, it was felt that an integration between Vis5D and METVIEW is highly desirable.

2.3.9 XY Plots

Many users felt that the current XY plots provided by METVIEW need to be improved. Two problems were especially outlined: the limited interface tools available for defining the axis composition and the curve displays, and the lack of flexibility for combining different information.

The combination of METVIEW with third-party packages (such as `PV-Wave` and `XMGR`) was proposed as a means of improving its XY plotting facilities.

2.3.10 Macro Generation from PlotWindow

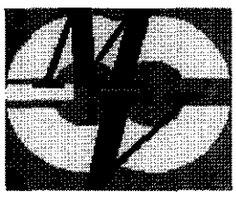
Some users stated that they felt that the facility of generating a macro from a plot window did not produce satisfactory results, as the macros produced were considered too complex for the majority of users. They suggested that this facility should provide simpler code, which the user could more easily modify.

2.4 Resume of Users Perspective

The following list indicates the most significant improvements and additions to `VisMod` which were requested by the consensus of METVIEW users. They are given on a priority list, which is based on two criteria: the users' demand and an expectation of what is realistic to expected, given the development constraints:

- Improvement on the graphics performance.
- More flexibilty for displaying results from applications (especially for non standard 2D maps).
- Better support for X-Y plots and ASCII data.
- Improved colour facilities.
- Consistency between batch and interactive processing.
- Consistency between normal visualisation and animation.
- Provision for inclusion of 3D visualisation software (Vis5D).
- Better suport for text and legend display, including more informative texts and continuous-tone legend support.
- Additional image processing functions.

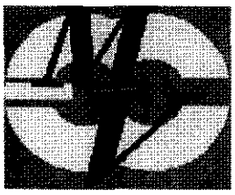
[Previous] [Next] [Contents]



METVIEW - Plot Module

3- System Design

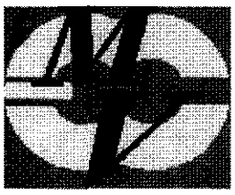
[Previous] [Next] [Contents]



METVIEW - Plot Module

3.1 Conceptual Design

[Previous] [Next] [Contents]



METVIEW - Plot Module

3.1 Conceptual Design

3.1.1 Introduction

The conceptual design for `PlotMod` is best understood by considering a simple example: suppose a user wants to display the same DataUnit (model analysis, temperature variable, 10 pressure levels) in three different perspectives: a 2D contour plot for a selected geographical area, a cross section and a vertical profile. He wants all output plotting areas to be related, so that the same layout could be applied to different data sets.

The user would also like an output layout similar to the one shown in Figure 3.1, where the topmost part of the plot window shows a vertical profile and a cross-section, and the lowermost part contains three of the possible 10 pressure levels. He would use a scrollbar (not shown on the figure) the display all 10 different pressure levels, as he does in the current `VisMod`.

This plot window organization, although simple to conceive, is impossible to be obtained in the current `VisMod`. In order to cater for this kind of layout, `PlotMod` will rely on some concepts:

- Pages: independent parts of the drawing area, where data is shown (the above figure has three pages, two in the top part and one in the bottom).
- Subpages: data-dependent drawing areas, which are the places where a page places its graphical output (in Figure 3.1, each of the top pages has one subpage, and the bottom area has 10 subpages, out of which 3 are currently visible).
- Views: each METVIEW application is associated to one view, which defines how the graphical output is produced. Each page is also associated to a view. In Figure 3.1, there are three views: the upper right page is mapped to a XZView (associated with the cross section applications), the upper left page, to a PZView (associated with applications such as Vertical Profile), and the lower page to a XYView (used for applications which display onto a XY geographical area, such as field contouring, image and observation plotting and colour wind).

These concepts are described in more detail in the next sections.

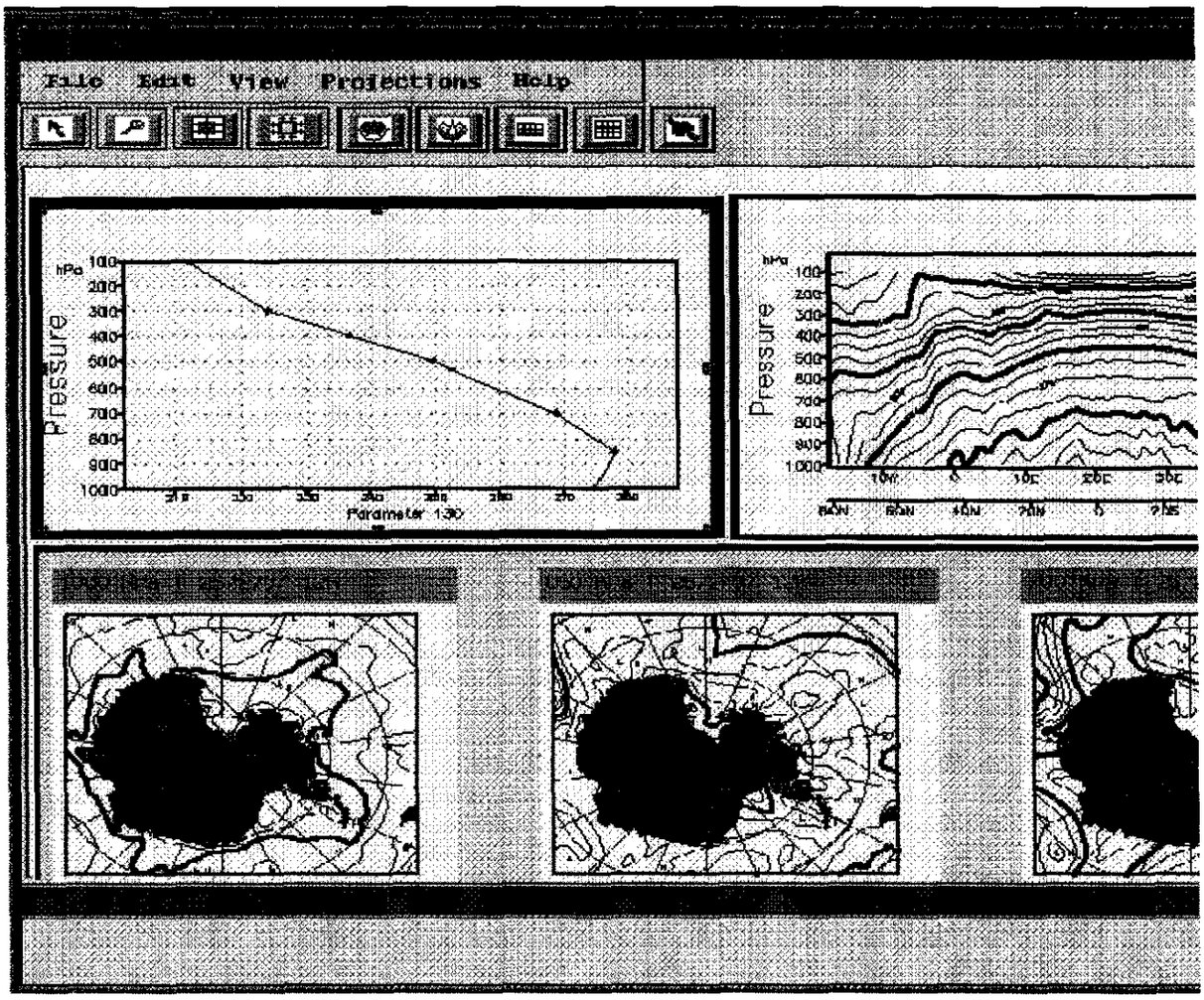
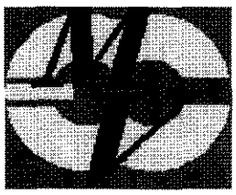


Figure 3.1 - Example of one data unit, different views

[Previous] [Next] [Contents]



METVIEW - Plot Module

3.1 Conceptual Design

3.1.2 Pages, Subpages and Superpages

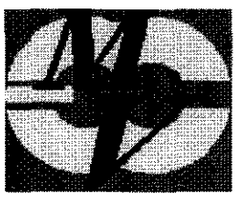
`PlotMod` displays meteorological graphics in *plot windows*. Each plot window will have its own user interface, and will be completely unrelated to other plot windows. In order to allow for different DataUnits to be displayed in one plot window and for the same DataUnit to be displayed in different views, `PlotMod` uses the concept of "pages". A *page* is the entity which can receive requests to display DataUnits (and associated VisDefs).

The example shown in the lower part of Figure 3.1 shows a case where a DataUnit contains fields for different pressure levels which are to be plotted separately, whithin the area allocated to a page. In order to support this behaviour, we need to introduce the concept of *subpages*. A *subpage* will contain a single plot of a meteorological variable (or a matching combination of meteorological variables). Subpages are data-dependent and are created by their parent pages according to matching rules (described below). Each subpage is associated to one or more drawing areas (or canvas), and has associated text and legend areas.

In graphical terms, a page corresponds to an area of the display surface (screen or paper), and a subpage will be a window which is contained completely inside it. From the point of view of the window manager (Motif), each subpage will be a separate drawable (X window or X pixmap), and will respond independently to events such as Expose and Resize.

Note that, whilst the actual number of subpages is data-dependent, not all subpages of a given page are displayed simultaneously. In a similar fashion to the current `VisMod`, the user will be able to decide how many subpages will be visible. A scrolling bar (similar to the current animation bar in `VisMod`) will allow the user to display all subpages. Details of establishing the visual layout are discussed in section 3.1.7.

[Previous] [Next] [Contents]



METVIEW - Plot Module

3.1 Conceptual Design

3.1.3 Pages and Views

One very important problem in METVIEW is the control of the visualisation output. In a way similar to desktop spreadsheet applications, each page in PlotMod is a associated to a *View*. A view is a particular way of presenting a meteorological variable, and each application is associated to a *View*. Table 3.1 below describes the views supported in METVIEW and the applications associated with it.

By default, when a page is created, it is not associated to any view. The user will be responsible for defining which view is associated to each page, and for establishing its parameters (e.g, area and projection in the case of XYView, scaling of axis in the case of Axis View). This behaviour contrasts with the current VisMod, where by default a plot window is created with an XYView (with a default area of the whole globe in cylindrical projection).

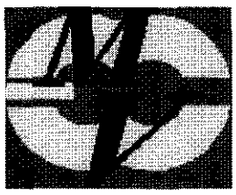
TABLE 3.1

VIEWS AND APPLICATIONS IN METVIEW

VIEW	Description	Parameters	Related Applications
MapView	A geographical area with coastlines, associated to a cartographical projection	1. Bounding box (in lat_long) 2. Projection	Field Contouring, Image Display, Obs Plotting, Wind Plotting, Colour Wind, Vertical Potential, Total Rain.
CrossSectionView	A cross section of the atmosphere, based on a straight line on lat/long coordinates.	1. Line on earth's surface (lin lat-long coordinates) 2. Initial and final levels 3. Vertical axis organization (linear/log)	Cross Section, Averages
TephiView	Two drawing areas with the special coordinate system used by the tephigram application	1. Minimum and maximum temperatures 2. Top and bottom pressures	Tephigram
VerticalProfileView	An axis where the horizontal coordinate is a parameter and the vertical coordinate is a height	1. Top and bottom pressures 2. Vertical axis organization (linear/log) 3. Point on earth's surface (lat/long)	Vertical Profile
TimeAxisView	An axis where the horizontal coordinate is a time measure, and the vertical coordinate is a parameter value	1. Vertical P Axis (scale, min and max values) 2. Horizontal T axis (initial and final time, increments) 3. Graph type (curve, bar chart, area). 4. Graph parameters: colour, thickness, line type and smoothing, symbol	Metgram
AxisView	A general axis where the user may establish what to use for horizontal and vertical coordinates	1. Horizontal and Vertical Axis (scale, min and max values, increments, colour, symbols) 2 . Graph type (curve, bar chart, area). 3. Graph parameters: colour, thickness, line type and smoothing, symbol	Curve

Medium-Range Weather Forecasts/ECMWF, Centro de Previsão de Tempo e Estudos Climáticos (CPTEC/INPE).

[Previous] [Next] [Contents]



METVIEW - Plot Module

3.1 Conceptual Design

3.1.4 Views and Applications

General Description

In this design, the applications will be separated from the views. After the user has associated a view with a page, this page will now be able to accept *Data Unit* drops and *Application* drops. *Applications should only provide data,* which will be displayed on a plot window which has a compatible view. Therefore, there are two possible ways of providing data to `PlotMod`:

- Dropping DataUnits into a plot window
- Dropping Applications into a plot window.

Case 1 - Dropping DataUnits into a Plot Window

When the user drops a data unit icon into the page, there are some possibilities:

- The page has no associated view. In this case, the page will be assigned to the default view (XY View), and the default application (Field Contouring) will be called to provide the necessary data.
- The page has a view, but no data. In this case, the matching rules apply, and the default application for that view is called to provide the necessary data.
- The page has a view, and data (associated with an application). In this case, the matching rules apply and the current application associated to the page will be called.
- The page already has an associated view which is incompatible with the data unit. The drop is then rejected.

Therefore, to obtain a normal field contouring, the procedure required would be:

- Define a XYView associated to a page (geographical area and projection).
- Drop a DataUnit and associated VisDef into the page. The contouring application will be called to produce the desired output.

Case 2- Dropping Applications into a Plot Window

When the user drops an application icon into the page, there are three possibilities:

- The page has no associated view. In this case, the page will be assigned to the deafult view associated with the application, and the application will be called to provide the necessary data.
- The page already has an associated view which is compatible with the application. In this case, the matching rules apply, and the application is called to provide the necessary data.

- The page already has an associated view which is incompatible with the application. The drop is then rejected.

The procedure for producing a cross-section would be:

- Define a XZView on a page of a plot window. This would be done by an interactive interface where the user chooses the geographical line in lat/long, the scaling associated to the Z axis.
- Create a new Cross-Section application in GenApp.
- Associate a DataUnit to this cross-section application.
- Drop the cross-section icon into the page. The cross-section application will be called, and the XZView parameters passed to it by `PlotMod` to produce the correct output.

Additional Notes on Applications and Views

The possibility of dropping both DataUnit and Application icons to one single page creates an assimetric situation, which may be the case of later problems. The alternative would be to create a new icon for the default application (Field Contouring), and to allow only applications (and no Data Units) to be dropped into a plot window. However, this could create a behaviour which is very different from the current `VisMod`, and might confuse current users.

METVIEW - Plot Module

3.1 Conceptual Design

3.1.5 Page Hierarchy

In figure 3.1, the same data is displayed in three different pages. Altough this could be achieved by means of the separate drops on the same DataUnit into each of the pages, `PlotMod` will provide a way to obtain the same result with only one drop. To that end, the plot window is organized as a page hierarchy, illustrated in Figure 3.2.

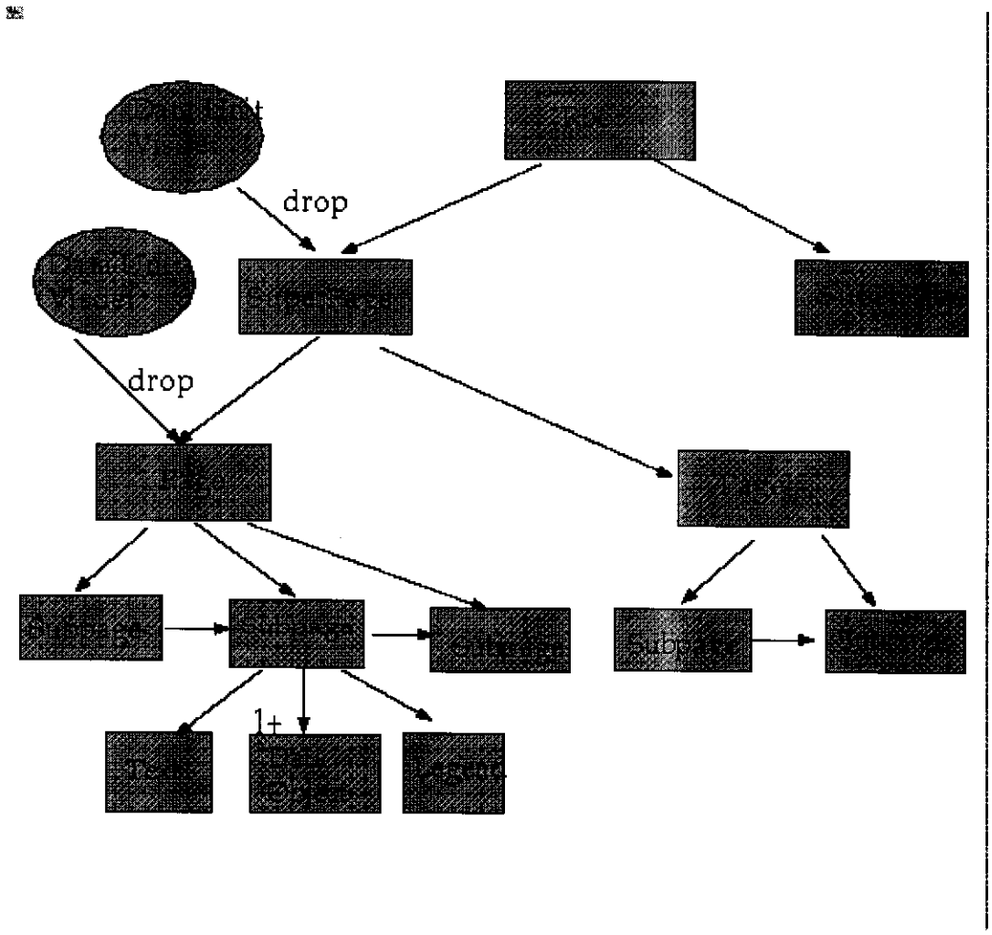


Figure 3.2 - Page Hierarchy for `PlotMod`.

The top-level node of the hierarchy is an abstract level, referred to as Root. The objective of this top-level node is to provide default descriptions for all elements of the page hierarchy (such as supplying the default view of a page, or the default coastlines for a subpage).

The next level on the page hierarchy is the instance of the plot window itself. This instance is referred to as a *Superpage*. Each superpage contains a number of pages (the third level). Data Units and

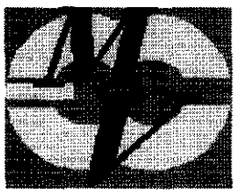
VisDef can be dropped by a user or be entered from a macro into either pages or superpages. DataUnits dropped at nodes which are not at the lowest level of the trees will be applied recursively down the tree, until they reach the lowermost level.

Each page has a number of children, called *Subpages.* The subpages will be created by their page parents, as required by the matching rules, and the user does not interact with the subpages directly, but can influence their behaviour by modifying the matching rules. To take a simple example, a GRIB file containing 10 different fields and dropped onto a page which does not contain any data will by default generate 10 subpages.

The leaves of the page hierarchy are the ones which effectively control the visual output, using the concept of a *Data Object*. A data object is the most atomic type of information handled by PlotMod, consisting usually of a single scalar or vector field or image, or a selected set of observations.

Text and Legend objects are also assigned separately to the Subpage, in order to allow for better control from the user.

[Previous] [Next] [Contents]



METVIEW - Plot Module

3.1 Conceptual Design

3.1.6 Matching Rules and Subpage Creation

The subpages are data-dependent, and are created by each page at the lower-most level of the tree, based on the DataUnits dropped onto it (or assigned to it by a macro command `plot`) and on "matching rules". These rules determine whether two meteorological variables are to be plotted on a single canvas or on different drawing areas. Matching restriction follows some general guidelines which will provide a predictable framework for METVIEW users, and some specific rules for each view.

General Matching Rules

- Rule 1.1 - "All is well that matches well"
  - All the attributes associated to a meteorological variable may be defined a criteria for matching, including: time, level, parameter, variable type and level type. Matching can be enabled or disabled by the user for all supported attributes.
  - By default, METVIEW will assume that if a user has requested to view two different data units on the same window, he has a good reason for it. Therefore, the default action will be to try to find a way to display these different data units together. Only when no sensible course of action is possible, will the user drop be rejected.
- Rule 1.2 - "Allow for non-synoptic hours"
  - A time tolerance will be applied to time matching, to allow for data collected at non-synoptic hours (such as images and observations) to be plotted together.
- Rule 1.3- "Different Data, Different Colours"
  - For all situations, the default behaviour when plotting two or more data units in the same window will be to use different colours for displaying them, to make comparison easier.
- Rule 1.4 - "Same Parameter, Same Axis"
  - For all axis plotting (such as Vertical Profile, Metgram) there are situations when two different data units are plotted. If the parameter is the same, the axis will be re-scaled to allow for the variation on the two data units.
- Rule 1.5 - "Different Parameters, Separate Axis, Same Scaling"
  - For axis plotting, there are cases when two different parameters are to be plotted together (such as 2m temperature and dew-point temperature). In this case, a second axis will be drawn, which will display the variation of the second parameter. In order to allow for consistency, both axis will have the same scale.

Matching Rules for Applications associated with a MapView

- Rule 2.1 - "One Drop, many fields"
  - When one DataUnit is dropped into a page, each field of a DataUnit is to be drawn on a different subpage.
- Rule 2.2 - "Two or More Drops"

- When two DataUnits are dropped into a page, individual fields from different DataUnits are plotted together if the matching criteria is satisfied.

Matching Rules for Applications associated with an CrossSectionView

- Rule 3.1 - "One Drop, One Time"
  - All data from one drop (contained in the its definition) are sent to the application program for processing. It is the responsibility of the application to verify whether the data is correctly defined for it.
- Rule 3.2 - "One Drop, Different Time-Steps"
  - By analogy with the case of the MapView, as many different subpages will be created as there are different time-steps.

- Rule 3.3 - "Two or More Drops"
  - `PlotMod` will accept two drops to be plotted together, provided that they statisfy the matching criteria.

Matching Rules for Applications associated with an VerticalProfileView

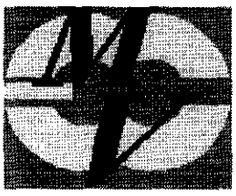
- Rule 4.1 - "One Drop"
  - If more than one parameter is contained in the drop, `PlotMod` will create one subpage per parameter.
  - If more than one timestep is contained in the drop, `PlotMod` will create one subpage per timestep.
  - After this filtering process, DataUnits are sent to the application program for processing. It is the responsibility of the application to verify whether the data is correctly defined for it.
- Rule 4.2 - "Two Drops, same Parameter"
  - By analogy with the MapView, if the parameter in the two DataUnits is the same, and other criteria match (e.g., level and time) the Data Unit will be plotted on the same subpage.
- Rule 4.3 - " Two Drops, different Parameter"
  - If two drops with different parameters are dropped in the same window, a second horizontal axis will be created (with a different colour) to display the values of the second parameters. Both axis are to have the same scale. The vertical axis is also re-scaled, as needed.

Matching Rules for Applications associated with TimeView (e.g, Metgram)

- Rule 5.1 - "One Drop"
  - If more than one parameter is contained in the drop, `PlotMod` will create one subpage per parameter. By definition of metgram, all time steps will be plotted together.
- Rule 5.2 - "Two Drops, Different Time Steps"
  - By analogy with the MapView, if the parameter in the two DataUnits is the same, and other criteria match (e.g., level and time) the Data Unit will be plotted on the same subpage.
  - By default, the time axis will be extended to allow for the user to view the two data units together (this is consistent with rule 1.1). For example, a metgram could contain a variable (e.g. 2T) generated by the T-4 forecast and valid for 10 days, plotted together with the same variable generated by the T-1 forecast.

- Rule 5.3 - "Two Drops, Different Parameters"
  - In accordance with Rules 1.1 and 1.5, when two different parameters are to be plotted, a second vertical axis will be created, to display the values of the second parameter.

[Previous] [Next] [Contents]



METVIEW - Plot Module

3.1 Conceptual Design

3.1.7 Layout Definition

The plot layout determines what is seen, not what is plotted. The layout definition consists in specifying the page hierarchy, in the terms of page and super-pages. In the initial version of PlotMod, the specification of the Plot Window will be done by an editor which will suport the page definition in the similar way as as the current "SuperPage" command, with the extension that pages can be embedded within other pages, and that a page could have visible subpages, specified in terms of rows and collumns.

Therefore, the textual description which is equivalent to Figure 3.1 would be:

```
window_height = 500
window_width = 500

profile_view = view (
    TYPE                    : PZVIEW,
    TOP_PRESSURE            : 1000,
    BOTTOM_PRESSURE         : 50,
    POINT                   : [-90,0])

ul_page = page( upper_left,
    view                    : profile_view,
    PAGE_X_LENGTH           : window_width/2.0,
    PAGE_Y_LENGTH           :  window_height/2.0)


cross_sect_view = view (
    TYPE                    : PZVIEW,
    TOP_PRESSURE            : 1000,
    BOTTOM_PRESSURE         : 50,
    LINE                    : [-75,0,-75,180]

ur_page = page (
    view                    : cross_sect_view
    PAGE_X_LENGTH           : window_width/2.0,
    PAGE_Y_LENGTH           :  window_height/2.0)

antartica = view (
    TYPE                    : "XYVIEW",
    MAP_PROJECTION          : "POLAR_STEREOGRAPHIC",
    MAP_HEMISPHERE          : "SOUTH",
    MAP_VERTICAL_LONGITUDE : 0,
    AREA                    :  [-52,45,-43,-130])

lower_page = page(
    view                    :  antartica,
    PAGE_X_LENGTH           :  window_width,
    PAGE_Y_LENGTH           :  window_height/2.0,
    N_ROWS                  :  1,
    N_COLS                  :  3)
```

```
plot_window = superpage(
    SUPER_PAGE_X_LENGTH     : window_width,
    SUPER_PAGE_Y_LENGTH     : window_height,
    plot_start              : "top",
    pages                   : [ul_page, ur_page, lower_page] )
```

The differences between this textual description and the one used currently in the superpage macros are:

- Inclusion of the `VIEW` directive, which creates new views.
- Association of a `view` to a `page`.

This textual description also allow for a backwards compatility to be attempted between `PlotMod` and the current macros which use the `superpage` directive.

In the interactive layout definition, a tree hierarchy will be shown to the user, displaying the current page hierarchy. For each page, the associated data units, views and visual definitions will be presented, allowing the user to interactively modify the contents of the page hierarchy.

[Previous] [Next] [Contents]

METVIEW - Visualisation and Plot Module

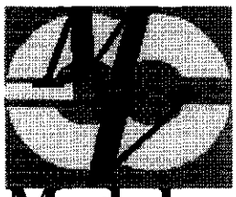
Design Document

3.2 Colour Choice

Introduction

This chapter is still under construction !

[Previous] [Next] [Contents]



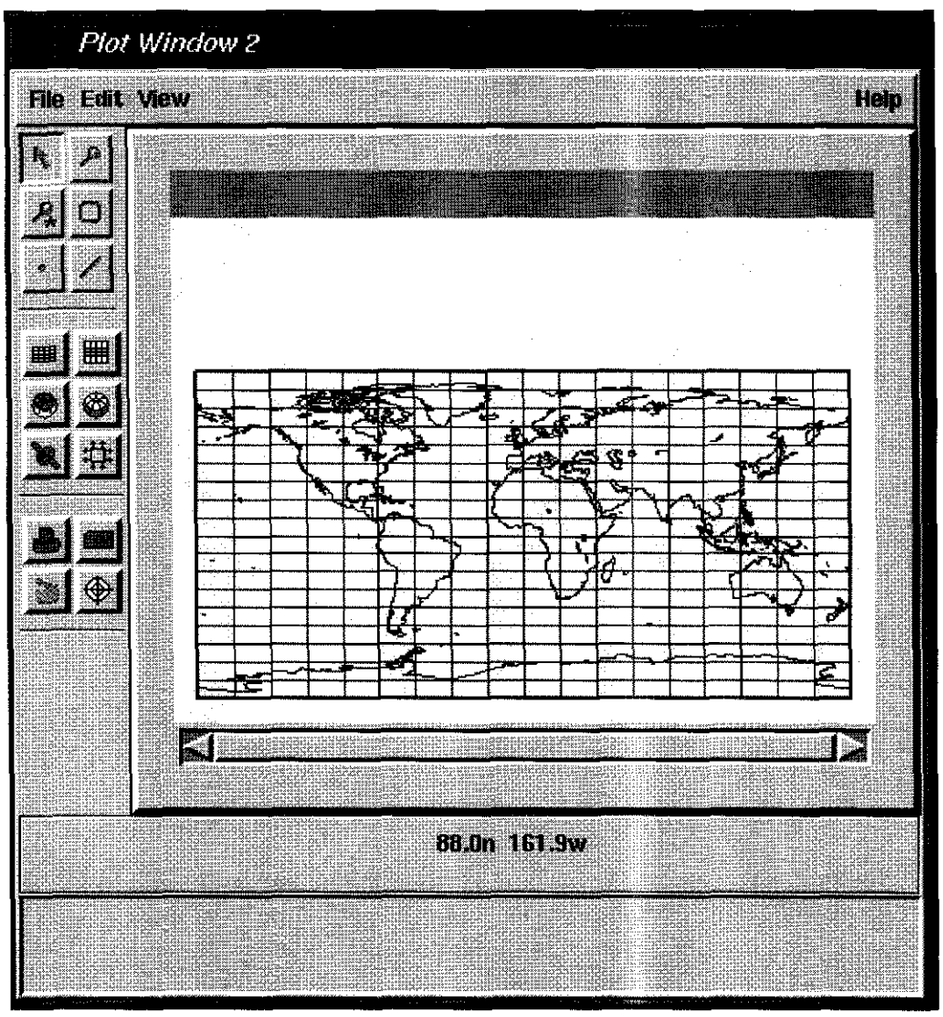
METVIEW - Visualisation and Plot Module

Design Document

3.3 New User Interface

3.3.1 Introduction

The default plot window interface for `VisMod` is shown below. The comments on specific items follow, with a view of generating a new user interface for use with `Plot Mod` .



3.3.2 Window Organization

As a whole, it was felt that the current interface does not make an optimal use of the screen for drawing maps. A lot of unused space is taked by the window where the icons are located and by the window which is used to display the cursor location.

A suggestion which has been made is to reorganize this window in such a way that:

- The icons appear on a window below the menu window, organized horizontally.
- The window which is used to display the cursor position is no longer used.
- The message area is made smaller, and a sliding bar is made available to loop through messages.

3.3.3 Icons

The issues most mentioned by the users regarding this interface were, as regards the icons, had to do with the fact that the icons do not correspond exactly to the actions performed by `VisMod`.

- The icons portrayed as lenses in reality correspond to two very different things: The magnifier icon and the zoom icon should be made different, since the second one effectively coorresponds to a new area definition.
- It was suggested that the new "zoom area" icon would be similar to , which would be put alongside the current icon , which corresponds to the magnification.
- There seemed to be an overlap between the icon used for showing grib values () and the one for selecting point locations (*). In practice, the two icons have to be combined. Since the default mode for the cursor (as selected by the icon) is a "point location" mode, the point icon was felt to be redundant. From that, it follows that the grib value output facility could be reduced to a single icon. Alternatively, the grib value would be shown in the top level window, alongside with the (lat, long) coordinates, as the default behaviour (which could be turned off).
- The majority of users had never used the "area select" facility (indicated by icon). This perhaps could be explained by the limited number of applications which make use of this facility.
- Due to the fact that having a screen preview is part of the normal working routine (partly because the current `VisMod` is not fully WYSIWYG, partly because the are inherent differences between screen and paper), it was requested that the Postscript preview icon is moved to the main Plot Window interface.

As a more general issue, there should be a menu equivalent to every icon on the `PlotMod` interface (the same comment applies to the `GenApp` interface), as prescribed by the "OSF/Motif Style Guide". Additionaly, each icon could be associated to a text "bubble" which briefly describes its content (similar to the Microsoft Word interface).

[Previous] [Next] [Contents]

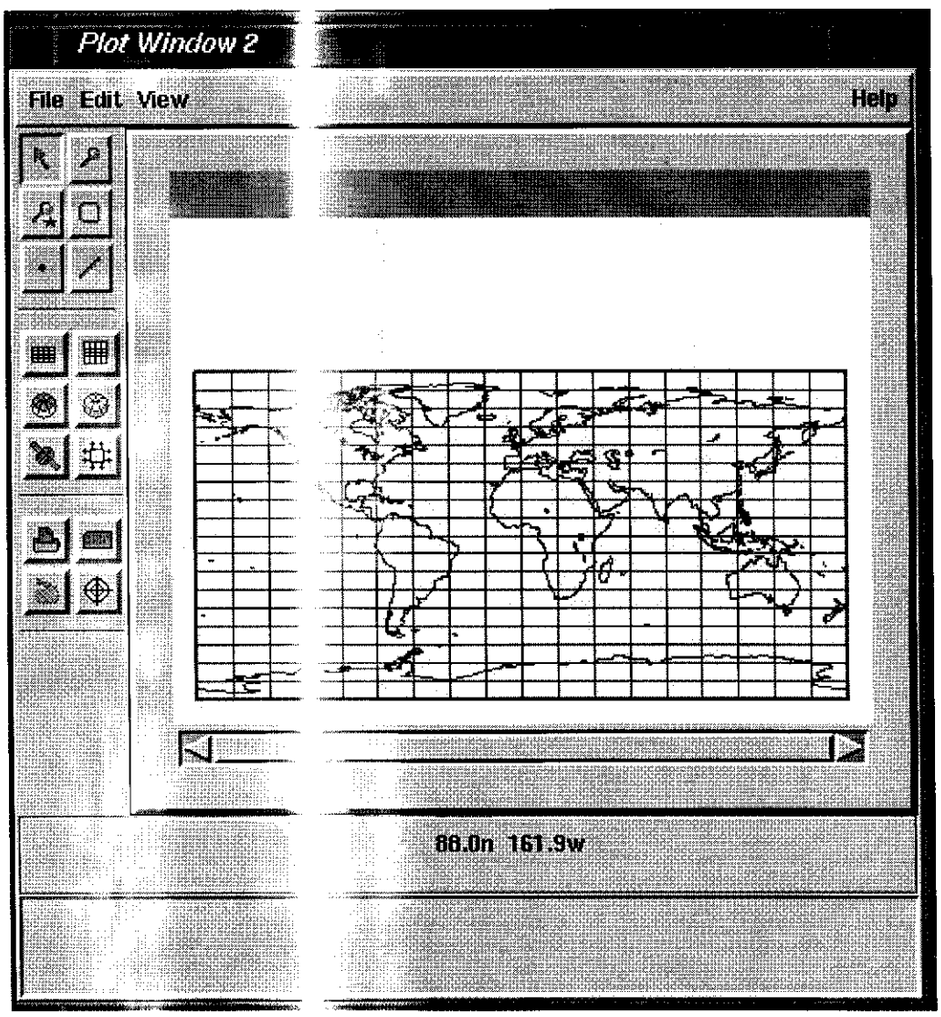
METVIEW - Visualisation and Plot Module

Design Documen:

3.3 New User Int :rface

3.3.1 Introduction

The default plot window interface fo VisMod is shown below. The comments on specific items follow, with a view of generating a :w user interface for use with Plot Mod .



3.3.2 Window Organiz .tion

As a whole, it was felt that the curr. interface does not make an optimal use of the screen for drawing maps. A lot of unused spa. taked by the window where the icons are located and by the window which is used to display th. rsor location.

A suggestion which has been made . reorganize this window in such a way that:

- The icons appear on a windo elow the menu window, organized horizontally.
- The window which is used to splay the cursor position is no longer used.
- The message area is made sm r, and a sliding bar is made available to loop through messages.

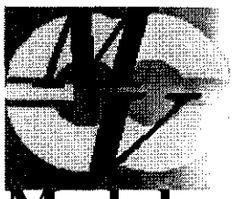
3.3.3 Icons

The issues most mentioned by the u regarding this interface were, as regards the icons, had to do with the fact that the icons do not c pond exactly to the actions performed by `VisMod`.

- The icons portrayed as lenses eality correspond to two very different things: The magnifier icon and the z om icon ould be made different, since the second one effectively coorresponds to a new area definition.
- It was suggested that the new oom area" icon would be similar to , which would be put alongside the current icon ich corresponds to the magnification.
- There seemed to be an overl tween the icon used for showing grib values () and the one for selecting point locations (In practice, the two icons have to be combined. Since the default mod. or the cursor (. lected by the icon) is a "point location" mode, the point icon was felt to be redundan. n that, it follows that the grib value output facility could be reduced to a single icon. Alte atively, the grib value would be shown in the top level window, alongside wi the (lat, long) dinates, as the default behaviour (which could be turned off).
- The majority of users had . sed the "area select" facility (indicated by icon). This perhaps could be explained by the limited number of applications which make use of this facility.
- Due to the fact that having . een preview is part of the normal working routine (partly because the current `VisMod` fully WYSIWYG, partly because the are inherent differences between scr n a. l paper). requested that the Postscript preview icon is moved to the main Plot Window interface.

As a more general . ue, there shou a menu equivalent to every icon on the `PlotMod` interface (the same comment applies to the interface), as prescribed by the "OSF/Motif Style Guide". Additionaly, each i n could be ass ed to a text "bubble" which briefly describes its content (similar to the Mic soft Word i).

[Previous] [Nex ntents]



METV EW - Visualisation and Plot Module

4. Syster ter es

Introducti

One important c n the de PlotMod is the need to preserve, to the maximum extent possible, compa ty with the c behaviour of VisMod, relative to the other METVIEW modu In other wo e impact of the substitution of VisMod by PlotMod for the other modules sho be as small p ssible. With that aim in perspective, the next three sections examine the inter between rent VisMod and the existing METVIEW modules.

4.1 METV Req Processed by PlotMod

Section 4. v es a gen scripton of the requests received by PlotMod and indicates the actions exp each si .

4.2 Detail mpl METVIEW Requests

Section 4. s a step p presentation of some typical requests processed by PlotMod.

4.3 Relatio tMo agics

Section 4.3 es an ar of the relation between PlotMod and Magics.

ET W - Visualisation and Plot Module

4.1 ME EW quests Processed by PlotMo

Introdu n

The primary me mmunic etween PlotMod and the other METVIEW modules is by means of reques ag-and- tions. The METVIEW modules which communicate with PlotMod are Me the mai nterface), macro (the macro player) and MagProc (the interface to MA ne dag- p actions by the user will also cause requests to be generated, as explained bel

There are six di tuations ch PlotMod processes and generates requests:

- Initializa ew plot
- Droppin onto a p dow by the user.
- Respons p action ted by MetviewUI.
- Interacti ch comn enerate by Macro.
- Editing a oviding ordinates for METVIEW applications).
- Messages s repor

Please note tha scussio the term "plot window" refers to a generic plotting device which may be a tive win a paper plot. Also, it may correspond to a normal plot window or to t notion erpage as defined by the SUPERPAGE command.

Initializatio plotti ons

Since plot win ndepend n each other, we shall consider the requests that generate a new plot wind tarting hese requests are produced in the following situations:

- The user lot win ther by double-clicking a an existing plot window icon, or by requesti ne to b from the icon menu (PLOTWINDOW).
- The user ne visu of a data unit (GRIB and BUFR).
- Metviev lotMod lise the result produced by an application (CLEAN).
- Macro in processi TWINDOW and SUPERPAGE).

In resume, ther hich init new plot window include the PLOTWINDOW and SUPERPAGE directives and , BUFR LEAN (when not preceded by DROP).

The reaction o to these s will be:

- A new p v is gen with the appropriate background.
- If there a ts asso the request, the matching rules will be applied to create the necessa for da lotted into.

- The data using ... fs given in the requests, and associated default values.
- A reply ... ne initi ... dule (`MetviewUI` or `macro`) giving the identifier associated to the pl ... . This i ... , called `DROP_ID`, is used for *all* further requests which deal with this ... ow. Thi ... holds both for interactive and batch use.

User Drag a ... › Actio ...

When a user d ... n onto a ... ndow, `PlotMod` responds to a drop by generating a request (`DROP_REQUES` ... rms Me ... about the identity of the plot window which has received the drop (this i ... e param ... OP_ID, `DATA_ID` and `VISDEF_ID`). The idea here is that `MetviewUI` ke ... the ic ... ed in the drag-and-drop operation and matches the icon's identity to the ... inform ... otMod.

MetviewUI ... e to a ... nd-Drop

After recieving ... tMod th ... indow's identification, MetviewUI will send a `DROP` request back to PlotM ... ontain t ... _ID corresponding to the desired plot window is followed by requests contai ... units (su ... RIB, `VECTOR_FIELD` or `MATRIX`) and/or visdefs (e.g: `PCONT`, `PCOAS` ... e data t ... been generated by an application, a `CLEAN` request is issued between the `DR` ... a requ ... h provides information about the application.

PlotMod's res ... onsist ... lowing steps:

- decode t ... g data.
- use the i ... iles in t ... letermine the appropriate setting for the plot window, includin ... ble crea ... ew subpages.
- plot the ...
- reply to ... indic ... t the request has been completed (or that an error has occurre ...

Macro Batc ... teract ... mmands

The communic ... een Pl ... d Macro follow the same pattern as the above-mentioned response to a ... p acti ... is understandable, since a macro is expected to be able to reproduce all ... rmed ... ely by the user. Therefore, the macro will genearate a `DROP` request (... a `DROP` ... owed by requests containing data units (such as `GRIB`, `VECTOR_FIELD` ... IX) and ... efs (e.g: `PCONT`, `PCOAST`).

Plot Mod's re ... e same ... : decode the data, match it against the current setting, plot the data, and i ... o of th ...

Providing M ... dinat ...

Many applica ... TVIEW ... oss-sections and vertical profile) require a geographical location (area ... t) as i ... he editing window associate with the application, the user may ask for a ... e whi ... es him with a geographical map (in cylindrical project) where he wou ... locati ... ctively.

This situation ... s to an ... NDOW request, which is sent by `MetviewUI` to `PlotMod`. This request i ... Mod t ... t a Map Editor Interface to the user, who makes his choice. PlotMod then ... letviev ... a `MESSAGE-INPUT` request which informs the desired coordinates.

Messages a · Info ı

Resume

The following nes the f t actions performed by `PlotMod`.

Situation	iption	Input Request	Action	Reply Requests
New P' Windo	on is -click lot wi uested	PLOTWINDOW, GRIB, UFR, SUPERPAGE, LEAN	Match Data, generate a new Plot Window and plot the result	REPLY-DROP (with DROP_ID)
User D	is mad ng plot w	(none)	Compute DROP_ID, DATA_ID, VISDEF_ID	DROP_REQUEST, DROP
New Data or VisDefs	s sent or ewUI t ng plot w	ROP-GRIB, ROP-CLEAN, ROP-PCONT (and similar ones)	Match data, change view (if necessary) and plot the result	REPLY-DROP
Map Coordi	inter: ewUI es a m n	NPUTWINDOW	Provide a map interface and obtain area, line or point coordinate	MESSAGE - INPUT
Messages an Status Information		ESSAGE-STATUS		MESSAGE-STATUS

Last Update 2 7 by C imara and Fernando Ii. Copyright 1997(c) European Centre for Medium-R ier Fo MWF.

ET N - Visualisation and Plot Modul

Design imc

4.2 Re be n PlotMod and GenApp

Introd n

T r is st r construction !

ET N - Visualisation and Plot Modu

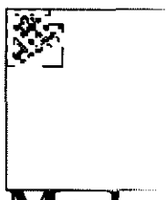
Desig ume

4.3 R be n PlotMod and MagProc

Introc n

T er is st r construction !

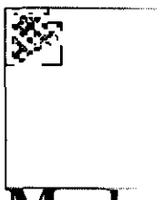
[Previous] [ontents]



ET W - Visualisation and Plot Modu

5. Sof Imp entation

[Previou ntents]



ET W - Visualisation and Plot Mod

5.1 S e In mentation Guidelines

Introd

Metview d s expecte ve over many years, and more features and functionaly are continuosl system. s perspective in mind, it is very important to use software implem es which maintenance and additions to `PlotMod`. Two principles are param g a good itation and developing flexible modules.

Docum

There are t ces of inf n on PlotMod: this document and the code itself. Instead of providing ment d the code, it was considered that the code description should be made p ment an hould evolve together. For that purpose, all the class headers are includ , with th ponding description of the class as a whole and of each individua class is d in terms of the CRC (Class-Responsibilty-Collaborators) model [E

The CRC us on vi ch class as responsible for a well-defined task, which it is *contracte* e clas perate with the one being described as referred to as the *collabora* ach vic t-oriented software development as the activity of design and impl istent sc ponents, which can be combined to form a complex system.

Imple Pri s

In the de tMod, n coding style has been pursued, in order to lighten the load of mainte es. So les have been applied to the maximum extent possible:

- A c style tation has been applied to all PlotMod classes.
- Th Tem ry has been used whenever applicable.
- The viour is izable dynamically, by means of external definition tables.
- The fined an C++ idoms like those described in Coplien (1992) and Gan) have fered, in order to improve code readability.

Codi

The codi 's as fo

- All met fined in uppercase, without underscores, e.g.

Pre orPa ::DataBase::NextVisDefByDataUnitId.
- All ith lo mes, but may include uppercase letters to indicate a
con in cur , defaultVisDef.
- All re ter y an underscore, as in dataUnitVisDefList_.
- Mc ide ir class members refer directly to the desired action, instead
of t qu .

Use of - Sta . Template Library

The STL - late Li ne of the most important recent additions to the tools
availbale nt. an extensive support for what has been known as *generic
progra m* s for pment of general algorithms and functions, which are
applica l ata ty STL includes support for lists, sets, vectors, associative
arrays (by emplate can be used for manipulation of user-defined
classes.

Futhermo been a s part of the new C++ standard.

Exter ed l iour

One of h TVI pment in general is the sheer diversity of requests, which
includ e atio g areas (such as PLOTWINDOW, SUPERPAGE), data (
includi ATI EOPOINTS), visual definitions (PCONT, PWIND), to
name l

This dive o gen rams which a significant part of the code is spent inquiring
the co a req hen performing an appropriate action. The code would look
like this:

```
if      and =      ')
        .
els     .Comm      'PCONT")
          els
...       re .       else)
```

In ord codi , we have attempted to concentrate most of the information

related to f requ tMod on an external file.

The co requ ssed by PlotMod and the actions associated to each request
are conta l file, *otModTable*, which is easily understandable and which
allows e the behaviour.

For exam m thi rding the SUPERPAGE command, indicates:

```
ass      AGE,
tion     e,
uilde    PageBuilder,
tch
```

The al cates UPERPAGE command is associated to the "Create" action,
which t wir that it uses the "SuperPageBuider" to build the tree
hierarc ning ed (since a superpage contains no data). In combination
with th " (as d elow), a significant economy on both code size and
complexit s le.

Pro lion otypes

The ba classe v. rocess the requests are coded using a C++ idom called "*progr rs*" [(92]. This idiom allows dinamic reconfiguration of the derive e used y in the MetviewUI (GenApp).

One o ms in opment is how to hide the dependency between the base class a For e us suppose that PlotMod has a base class Builder which takes t win hich has to deal with various types of commands.

A "co ld be :

```
 == S
perPa      (tree);
mmand      INDOW)
ll P       uilder (tree);
 (co       RIB)
ll Gr      (tree);
so on
```

The " laces on the client application side simply by:

```
= Bu      ke (command);
(com      e);
```

The "s erived ch implements the specific builder for each type of comm "regis first main program (`main.cc`) is performed. The program then proce st ap pe of builder (which is performed by the `Builder::Make` meth), the nportant benefits:

- s with sses (such as Builder), and all the subclass information
- can b namically, without the need to recompile the base class or

Last U by G nara. Copyright 1997(c) European Centre for Medi oreca F.

ET V - Visualisation and Plot

M

5.2 ntir Page Hierarchy

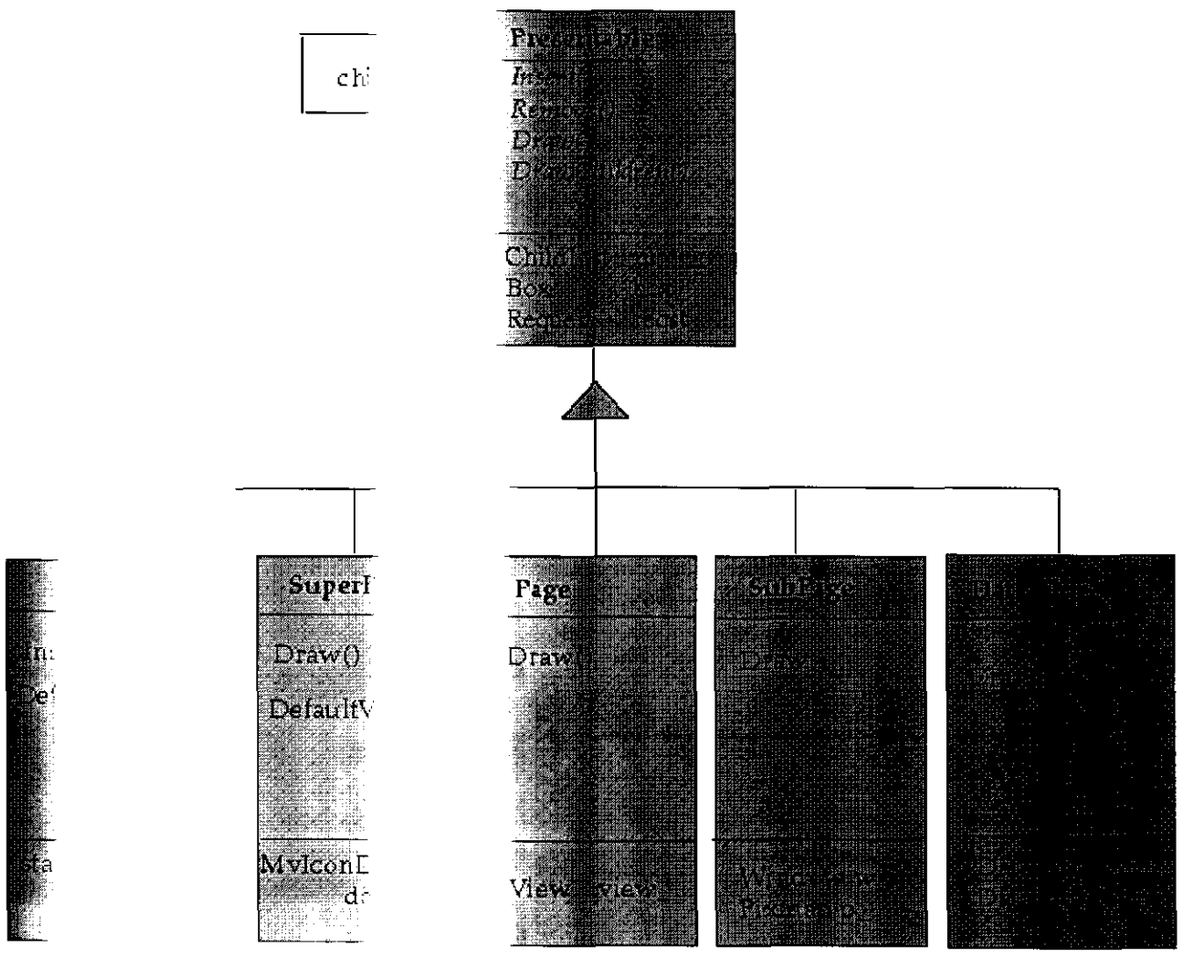
Given archy is f PlotMod, it is most important to devise an adequate
repres tially, it i o consider that the page hierachy described in section
3.1.5 e, which co cts of different types. From a software design point of
view, e useful to objects (pages, subpages, views and data units)
unifo achieve gr and code cleanliness, PlotMod relies on a general
abstra ve will call *ble*. This class represents *both* the bottom-level structures
of the lata units a and their containers (pages and subpages).

The l ncapsulat methods for inserting, removing and drawing objects on
the pa for obtai ents and children on the page hierarchy tree. Each of
these ed as an a ction (in C++ terms, a virtual method), which will be
imple rete funct of its subclasses.

In ter terns", th le class is an example of the Composite pattern. A typical
use o ttern is for g objects into tree structures, such as a UNIX file system.
A UN y contain f other directories.

A bri each funct resentable class follows:

- ation will d bject and its children. For a subpage, this operation might
  le on the ou and call a Draw operation for all its children. For a
  ill require aphics engine to perform the required graphics
  ouring, fo .
- emove op eate new branches and leaves it the tree. Note that only the
  ges can in nove their children. This operation is defined at a higher
  sentation asons of ensuring greater transparency on the resulting
  r to the " rn" book (page 167) for a specific discussion on the



Ther iderations v is pattern for representing the `PlotMod` page hierarchy:

- ructures hav ype of container object (such as a directory in the em). In Plo are two types of containers: pages and subpages.
- objects are *primitives*: DataObjects, Text and Legend, since these are nits which
- t class is d the notion of METVIEW's DataUnit. In METVIEW, a ective y a pically a set of fields from the same forecast). In ject o the lass corresponds to one single field or image, or to a set s which ha ted during a specified period. The DataObject class leseri io eteorological data types, such as fields, images and The descrip only the necessary information for the graphics erate re
- egend obje idered to be independent entities from the DataObject, n has to inc ons to allow the user to modify their contents and

Each ierarchy s ore the information stricly necessary for its operation. This da ee n d clarity of the code. Therefore, when a command is issued at an ge ie ed ited information will descend the branches of the tree invol ch bj e v ie applicable operation.

Two y to c ar f

- ving a ti on to the entire plot window. This will amount to asking

topmost le [illegible] tself. In turn, the page will ask all its children to draw [illegible] a subp[illegible] [illegible]l, it in turn will send a drawing message to its children [illegible]mitive objec[illegible] [illegible]ataObjects). Each DataObject will then be linked to a [illegible]e (for exam[illegible] [illegible]S) which performs the actual drawing. The relation of [illegible]t to the gra[illegible] [illegible]y is explained in detail below.

- [illegible]DataUnit (fo[illegible] [illegible] set of fields) is inserted at a branch of the page [illegible] DataUnit [illegible] [illegible]nwards the tree. When it reaches a subpage, a matching [illegible] to det[illegible] [illegible] element of this Data Unit (i.e., any field) matches the [illegible] insert [illegible]page. If they match, a new child of that subpage is born. [illegible] fields w[illegible] [illegible]ed as children of new subpages.

There[illegible] general term[illegible] [illegible]ns on the implementation of any operation is delayed until [illegible] [illegible]nt. [illegible]dea [illegible]bject which performs the operation will know exactly how i[illegible]

[Previ] [Contents

METV - Visualisation and Plot Mo

5.3 ta Base Organizing Data Units and Vis

Mot

The ce ita Units and V nition (called *visdefs* for short) are quite central to the whole W eration, e concepts and their relationship are by no means obviou

The ba W sign e the output from a request to a meteorological data, whic a l ge num nt scalar and vector fields, for different time periods, parame els. Therefor taunit is actually misleading (it would be perhaps be more a c it a *dat val),* but it has been largely accepted by the user commu

A visu (or isdef) is se meaning in METVIEW is much closer to its semantic conten isd is used W to indicate a set of directives used for graphical prese eo g ml d

The r tw n data u ef is also quite complex. Many situations are possible, inclu i

- ay r nt isdefs, as in the case of one field being plotted with es bl a lues in red.
- d ay p n one data unit, as in the case where a superpage nt in le is used to display fields originating from separate r ati s (uch s not possible in the current VisMod, but is a requirement

An a s r o efs in the page hierarchy of PlotMod. The obvious solut e d o is a part of a data unit) would store all visdefs associated with r h is on would be placed in the leaves of the tree hierarchy desc 5. Su h ot feasible, however. Not only unecessary duplication of infor oc l t omplicate the implementation of the main user interface.

Da I S ons

In ord ie a it n PlotMod, two general approaches are possible. One is to co it l s al classes, and to create sub-classes which inherit from them u e e different type of entity (for example, Bac s th c `WindVisDefs` for wind plotting definitions). Such an appr a l e nilar classes, and to a large amount of context switching on tl

The (is oi id units and visdef as METVIEW interface icons, which
respc ni c y y or through a macro program). Such is the approach
taken

In Pl le iv its both entities. Each icon contains a copy of the
origi t\ e s a PCONT request), so that all relevant information can
allwa l f i rder to simplify operations, the MvIcon class is reference
coun c o t to the same memory positions. The deletion of an
objec vl o eleted. In PlotMod, the reference counting is
impl tl l mming idiom described in Coplien [1992].

Each is a ich is used to retrieve information about it from a data
base il s l.

En i]ag

Give ty re data units and visdefs, it is more appropriate to resort to
data c uta base entity-relationship terms, *data units* and *visdefs*
are *e* ar / illustrated by the Figure 5.3.1.

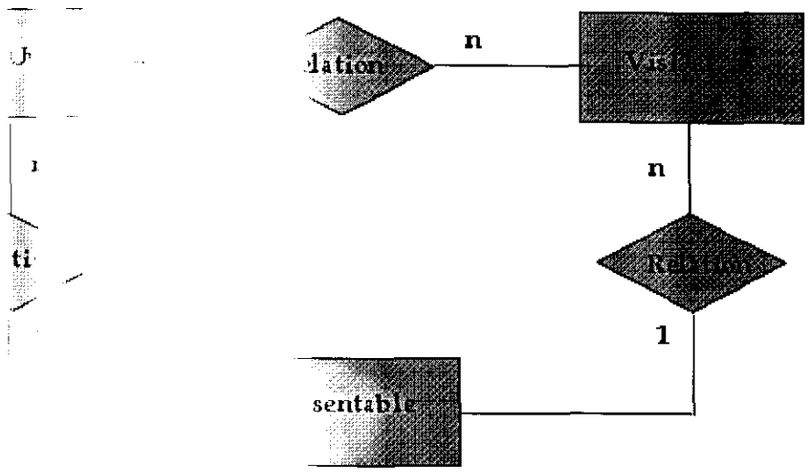
Ther re e visdefs in a data base, and to provide various
alter e n e used in different circunstances. In this approach, for
each ol g e a single, centralized place where the information
abou oc ti

This (al ui hip of all presentables and visual definitions which are
not a y a are used to display many data units at one. This would
provi a ec elds.

Ther l ill store:

- c s all visdefs (implemented as STL *lists*). The m:n
  e sdefs is implemented as an STL *multimap*.
- i ples and the visdefs which are associated to a
  i unit, implemented as an STL *multimap*.

In re: e: t of the `MvIconDataBase` class, which provides all the
infor h: which are used on all levels of that superpage.



Figur)(isdefs and presentables.

[Prev]

V - Visualisation and Plot

Mo

5.4 I sts

Ove

The ar s s associated with PlotMod, described in Section 4.1, has led to n actions which are performed:

- s and superpages.
- units and visdef into plot windows.
- contents).
- ages.

Each PlotMod class (`CreateAction, DropAction, Clos tAction`), which is called by the main METVIEW module depen est.

Dra

Out o important, as they lead to the generation of graphical outpu ions, and will be examined here in more detail.

In the sists of four main phases:

- window's page tree hierarchy.
- e device driver.
- tree hierarchy.

As ar ntaining a data unit is requested to be "visualised" by the us GRIB request. PlotMod will open a new superpage, then deco nd subpages are required to store it. After that, the grap actual drawing.

Taki led to implement the drawing actions in PlotMod to reflec classes are part of PlotMod, implemented as abstract base

- s `PlotWindowBuilder, SuperPageBuilder and`
- as `MapViewMatcher` and `VertProfViewMatcher`.
- s `Magics` and `Vis5D`.
- OpenGL.

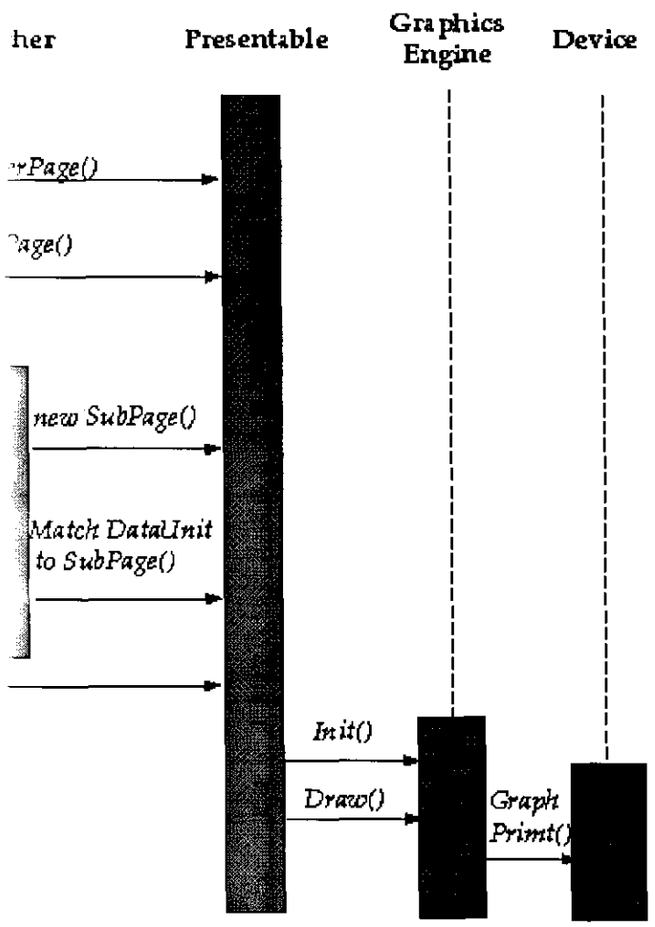
Int

The i
for tl
diagr

ted on figure 5.4.1, which shows the interaction diagram
itant cases. In essence, the flow of actions shown by the

- dAction object
- e request and calls a Builder, to build the new
- acther object matches the data units and visdefs data in the plot window.
- drawing message is sent from the PlotModAction age hierarchy. In turn, this branch (a page or a lity of drawing itself to a Graphics Engine. Vis5D) will contain functions for visualisation of t section) and will call graphical primitives from the
- hical primitives in XWindow, OpenGL and Postcript

Req

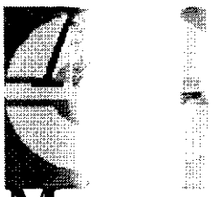


Last U

ra. Copyright 1997(c) European Centre for

Medi

[Previ



Mo e - Visualisation and Plot

5.5 t lowing Different Graphics En s L Drivers

The c 'T graphics engine, and MAGICS will continue to be the main s sc

PlotM ng fferent graphics engines, including the Vis5D visuali oft is, the `GraphicsEngine` class encapsulates the general behavi ge refore, the other PlotMod class deal with this abstract class, w fact that the graphics engine is MAGICS, Vis5D or anothe

Last l 5 J ra. Copyright 1997(c) European Centre for Medi e V

[Previ xt ontents]

IET EW - Visualisation and Plot

Mo)

6. C h ons d Acknowledgments

Co s

Ac vi gme s

The au kn w e important contributions to this conceptual design:

- hierarc em 3.1.5) is by Baudoin Raoult, who has given invaluable hole of ftware design.
- n App ns and Views (item 3.1.4) and the matching rules (item 3.1.6) discuss vith Vesa Karhila and Jens Daabeck.

er is s nder construction !