

Metview update

Fernando Li, ECMWF

Abstract

Metview is a meteorological data visualisation and processing tool. It was designed as a flexible, modular and extendible system with the ultimate aim of providing a complete working environment for the operational and research meteorologist. This paper describes the latest Metview developments, including a first attempt to integrate it with Magics++, the ECMWF's new plotting library. This experiment combines the existing facilities of Metview with the new ones provided by Magics++, building a framework for the easy design of portable XML-based (MagML) plot templates. The results confirmed the benefits of integrating Magics++ with Metview, one of which is improved interactivity.

1 Introduction

Metview is an interactive meteorological application designed to access, manipulate and visualise meteorological data on Unix workstations (*Daabeck et al. 1997*). It aims to provide a complete working environment for the operational and research meteorologist, by providing powerful data management, visualisation and processing tools. Figure 1.1 illustrates the data and work flow in Metview. In this example, MARS, the ECMWF data archival, is shown as the data source, but others may be used.

Metview can be seen as a meteorological desktop plotting package, thanks to its WYSIWYG (What You See Is What You Get) visualisation, but it is also a powerful meteorological data processing tool thanks to its Macro language, and it can be used for routine production of meteorological charts in an operational environment.

Metview was developed as part of a co-operative project between ECMWF and INPE/CPTEC (Brazilian National Institute for Space Research/Centre for Weather Forecasts and Climate Studies), with assistance from Meteo-France.

The aim of this paper is to give an update on recent developments in Metview. Section 2 presents an overview of the main features of Metview. Section 3 describes some of the latest Metview developments, most of which have already been incorporated in the current Metview export version. Section 4 discusses plans for the development of Metview and Section 5 presents the final notes.

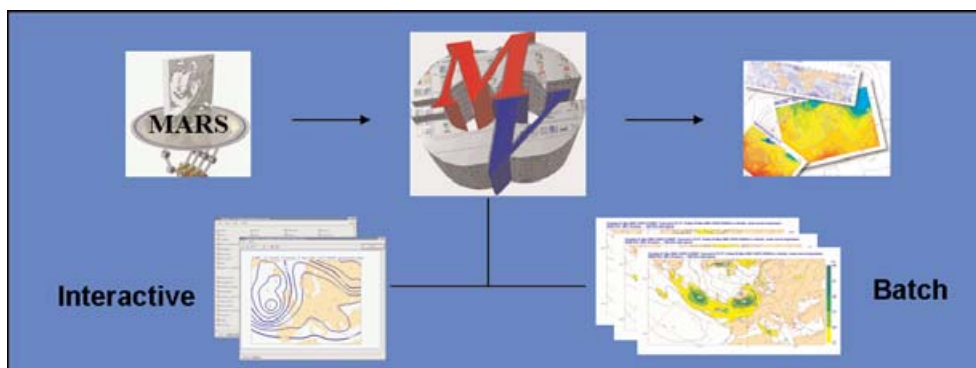


Fig. 1.1 The Metview working environment

2 Main features

Metview offers an extensive range of features that allow a meteorologist to use it as a complete working environment. These include not only an easy-to-use user interface and a powerful visualisation module, but also an extensive range of pre-defined applications, such as Cross Section and Data Coverage, and built-in utilities, including Metview Mail to send and Metview News to receive information.

Metview has a modular architecture and was conceived as a fully distributed system, where modules can run on different machines (Figure 2.1). It is based on well-known software packages, such as OpenGL and Motif libraries. Moreover, Metview was designed for the UNIX environment and is highly portable within the UNIX world, including some of the most popular Linux distributions, such as SuSe and RedHat.

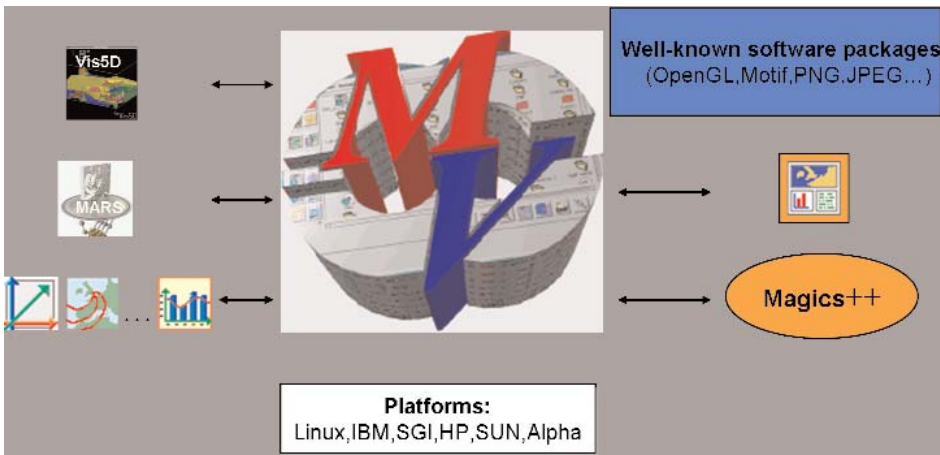


Fig. 2.1 Metview - Open and portable design

Thanks to its open design architecture, Metview is easily modifiable and extendable. In other words, it is simple to add new modules and features. Moreover, modules and libraries are configurable, e.g., users are able to select the Metview applications to be installed. In the near future, a new visualisation module will be added to take advantage of all facilities offered by Magics++, the ECMWF's new plotting library (Lamy-Thépaut, 2003).

Metview comes with an easy, powerful, meteorologically oriented Macro language, well suited for the most complex calculations with meteorological data (Karhila, 2003). The Macro language can describe very complex sequences of actions, particularly if the flow of action is conditional. It aims to be as simple as a script language and as powerful as a modern computer language. The Macro language extends the use of Metview into an operational environment as it enables a user to write complex scripts that may be run with any desired frequency.

An extensive list of well-documented Macro functions is available (ECMWF, 2006a). To create Macro programs users can select their favourite editor. In particular, the NEdit editor was customised to add several productivity features into Macro editing (ECMWF, 2006b). This includes macro syntax highlighting and code templates.

The fundamental unit of Metview is the icon. Figure 2.2 shows a typical Metview user interface. It is full of different icons, representing for instance data requests, plotting specifications and macro programs. All Metview interactive work is carried out through operations on icons. In fact, the basic Metview principles are: a) everything in Metview is an icon, b) every Metview task is a sequence of actions on icons. The main advantages of this concept are: a) easy to learn, b) no need to remember commands, c) intuitive to use (a picture is worth 1000 words).

Metview can also handle a variety of data formats, including WMO GRIB for numerical model fields and satellite images, WMO BUFR for observations, ASCII matrices of lat-long gridded data, ASCII lists of irregularly spaced data points (Geopoints) and NetCDF for data other than model fields and point data (e.g. cross sections). For both operational and research use it is frequently required to produce plots with more than one data unit overlaid, each requiring its own visual definition. Metview provides a rich set of visualisation attributes, such as contour shading, line thickness, density of observations and colour schemes. Although users are able to customise this wide-ranging choice of visual definitions, sensible defaults are assigned to enable the production of good quality plots, with minimum changes to the default values.

Other main features include a powerful and flexible visualisation module, where the ability to display different types of plots in any arbitrary layout is one of its most striking capabilities.

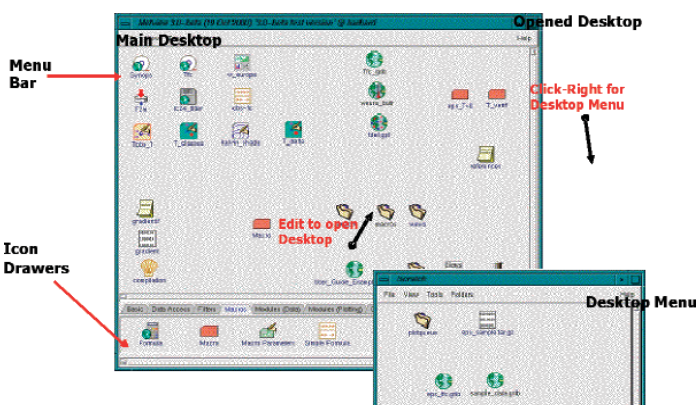


Fig. 2.2 The Metview User Interface

3 Latest Metview developments

Since its first release in 1993, several Metview modules have been revised to improve functionality and to cope with new data types. Internally, extensive rewriting and simplification of the code have taken place to make Metview easier to extend and maintain. Moreover, numerous new modules and functions have been developed to handle frequent new user requirements.

Among the new resources that have been recently incorporated into Metview are the production of Hovmøller diagrams, Tropical Cyclone Tracks, a Time Series application and new Macro facilities. A brief description of these new resources is presented below. Complete documentation can be found at the following web address:

www.ecmwf.int/publications/manuals/metview/index.html#news.

The Hovmøller application displays a two-dimensional graph with latitude as one axis and time as the other (Figure 3.1a). It generates a diagram along a specified arbitrary transect line or a rectangular area. If an input line is given, point values for each field are interpolated along the transect line, with a spacing consistent with the resolution of the input GRIB data. If a rectangular area is given, average values for each field are taken along the North-South or East-West direction.

The Tropical Cyclone Tracks application provides both deterministic and probabilistic information on the movement and intensity of individual tropical cyclones (Figure 3.1b).

The Time Series application plots a time series of a given parameter for a specified location (Figure 3.1c). The location can be defined by arbitrary geographical coordinates, or by choosing a station name from the WMO station database. The input parameter can be selected from GRIB, Geopoints or BUFR data.

The ability to inline functions written in Fortran/90, C, C++ and shell scripts within macros extends immensely the scope of the Macro language. Now there is no need to separately compile a routine; this is done automatically when the macro is run, using compiler settings that are consistent with the Metview installation. As a consequence, a macro which incorporates code from another language can be easily executed on machines with different compiler versions.

Moreover, it is also possible to run inline functions under a debugger process. When the macro is executed, the source code of these functions will be visible in a debugger program selected by the user. This facility will certainly help the integration of routines written in another language to a Macro program. In addition, new log flags are available which provide better control of the amount and content of the output information received from Metview components. It had become apparent that in an operational environment the original log flag usually provides too much information. Because of this, some important messages were difficult to spot. Among the new log flags are *-qlog*, which only prints error messages (no warnings), and *-mlog*, which only prints MARS retrieval messages.

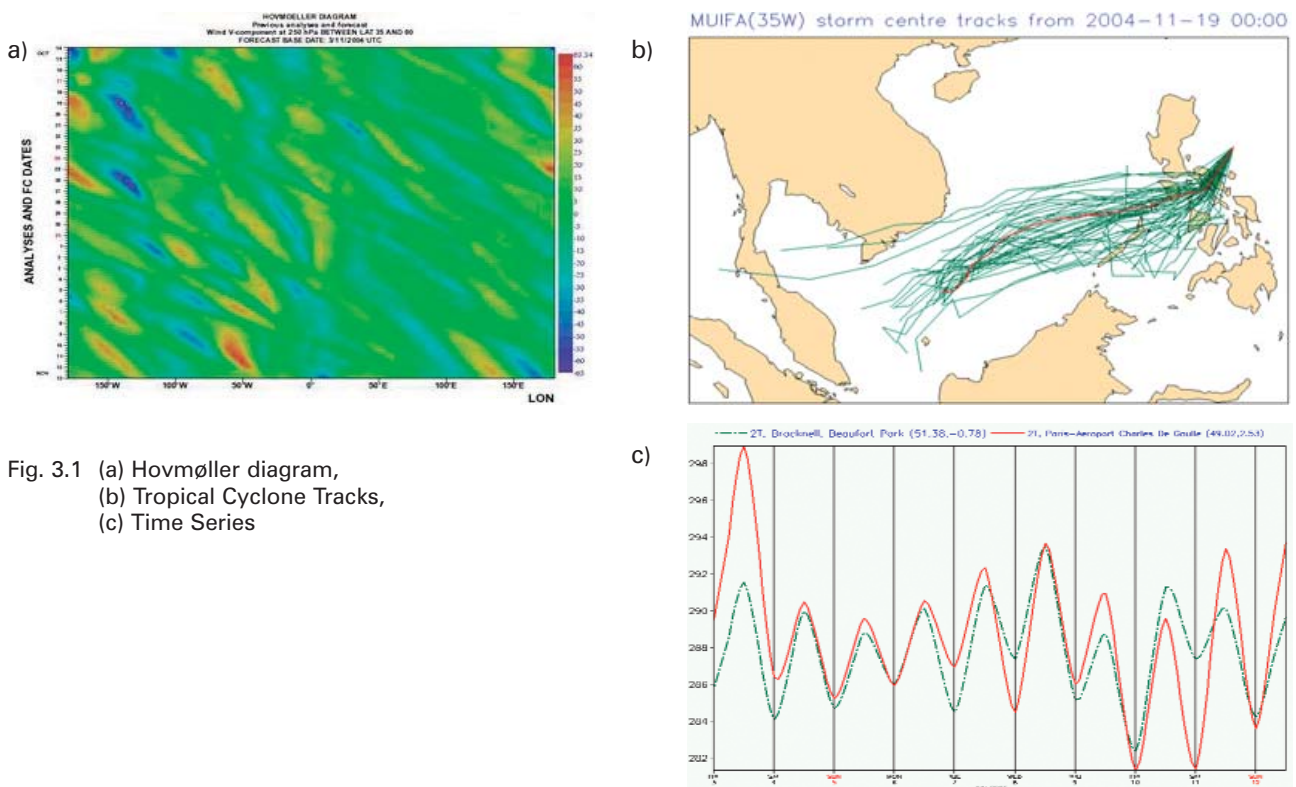


Fig. 3.1 (a) Hovmøller diagram,
(b) Tropical Cyclone Tracks,
(c) Time Series

4 Future plans

Metview is now a mature meteorological data processing and plotting package. However, there is a continuing routine effort to incorporate new facilities into it. This is a never-ending process, which includes constant improvements to handle new data types, new requirements from the meteorological community and new software facilities. These routine efforts are part of the main basis for the continuing usage of Metview - "making the processing and visualisation of large and complex data sets as easy and simplified as possible" (Bonifacio, 1999).

Currently, to install Metview, some basic working experience with the Unix operating system, Unix tools and programming on Unix is required. This is because a) Metview runs on Unix workstations, b) Metview is distributed in source code format and c) Metview requires some additional software to be installed. To simplify the whole installation procedure, the use of the GNU configure utility is being considered, since in general packages configured with this tool are straight-forward to install.

Another planned task is the integration of Magics++ into Metview. Magics++ is currently under development and will be the latest version of the ECMWF's meteorological graphics library MAGICS. As a consequence, a new visualisation module will be developed to take advantage of all the benefits provided by Magics++. These benefits include a) improvements in user interactivity, such as legend/text handling, b) the ability to dynamically toggle certain graphical objects, such as contour labels and maxima/minima, and c) querying the properties of graphical objects displayed on screen.

In addition, an improved communication mechanism between Magics++ and Metview is also expected. Both software packages are written in the C++ language, while the present MAGICS uses the Fortran language. Taking advantage of an object oriented language, the internal communication system will be based on objects instead of the current metadata methodology. The initial advantages of this new approach will be straightforward error/warning/text messages handling and simpler program code, which implies higher robustness and better maintainability.

In a first attempt to integrate Metview and Magics++, an experiment was built which combines Metview's current facilities (such as open design and modularity) with the new ones provided by Magics++ (such as the improved communication mechanism, user interactivity and the MagML interpreter), building a framework for the execution of interactive user tasks. MagML is a new facility in Magics++ for defining plots using an XML (Extensible Markup Language) description. The experiment uses PlotMod (the Metview's visualisation module) as a user interface to MagML plot descriptions.

Figure 4.1 shows two examples of this prototype. Basically, the following procedure was applied:

- PlotMod was modified by adding two new buttons: a point selection and an area selection.
- The list of operations attached to each button is read from a configuration file customised by the user. Users create MagML jobs and register them in their own Metview configuration file. This facility takes Metview to a new level of interactivity. Users are now able to define their own applications which require input from the map visualised on screen, and test them immediately.
- If a user chooses an application, for instance, to create an Epsgram (Figure 4.1a), and selects a geographical point location, the associated MagML job will be executed and the result will be presented on screen or on another output device.
- The same procedure applies for the area selection example (Figure 4.1b). In this example, a MagML job, called Amsub, will be executed which filters out observation data from ODB (Observation Data Base developed at ECMWF), given a selected geographical area.

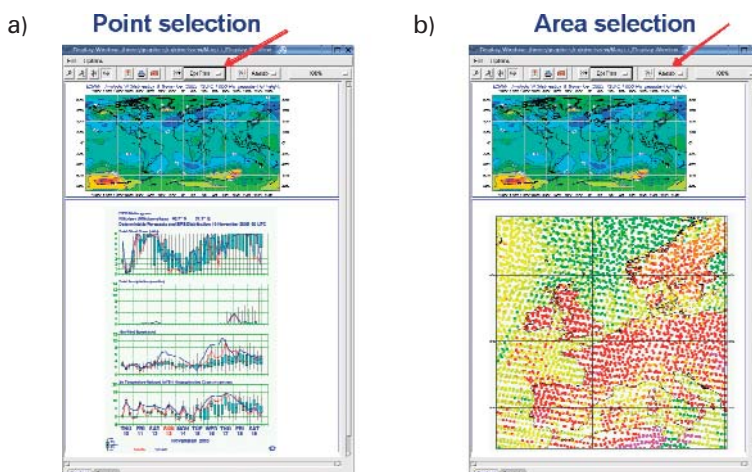


Fig. 4.1 Integration of Metview and Magics++: the use of PlotMod and MagML facilities

This experiment can be further expanded by building a framework for the easy design of portable MagML plot templates (Figure 4.2). These templates can be shared, for example, between Metview and other applications, such as the under-construction Plot-on-demand software, which produces on-the-fly web charts using the MagML language. At the moment, the plot templates are created only by the user. In the future, in order to populate this container of templates, it will be possible to create them using, for instance, a) a MagML editor, equivalent to the Macro editor, where facilities like dropping icon definitions (where the code is automatically generated based on the icon contents) and templates will be available; b) a MagML generator, similar to the Generate Macro tool, creates a template from the contents of a map visualised on the screen; c) a Macro to MagML tool to convert existing Macros to the MagML language.

5 Conclusion

Metview has been installed in more than 50 organisations and distributed in more than 20 countries.

Since its first release in 1993 (V1.0), Metview has gone through a number of operational versions. Metview V2.0, 1998, introduced a flexible new visualisation module based on the OpenGL library (Open Graphics Library). Metview V3.0, 2000, presented a revised main User Interface with improved functionality, where user customisation was the main priority.

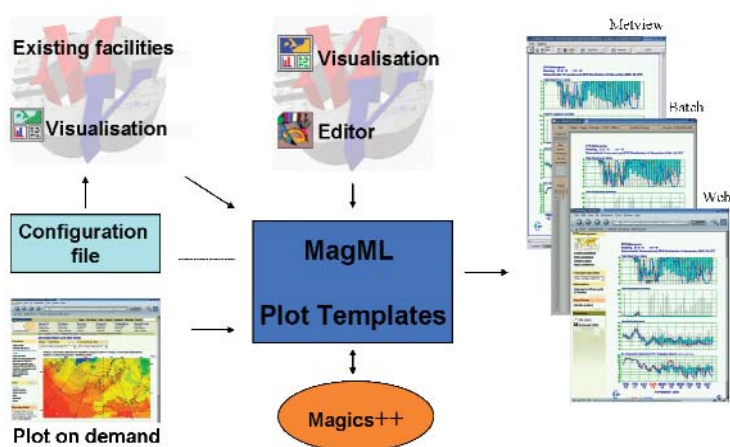


Fig. 4.2 Integration of Metview and Magics++: MagML plot templates

Metview V4.0 will be the next target. It will bring together a new visualisation module based on a new graphics engine. The new facilities described earlier will greatly enhance the flexibility of Metview and add many powerful benefits for its users.

References

- Bonifácio, R., 1999. METVIEW - Meteorological data visualisation and processing software at ECMWF. *ECMWF Newsletter*, number 86, pp. 6-18.
- Daabeck, J.; Raoult, B.; Camara, G., 1997. The Metview Application at ECMWF. In: 13th International Conference on Interactive Information and Processing Systems for Meteorology, Oceanography and Hidrology, Long Beach, EUA, February 2-7. Proceedings, American Meteorological Society, pp. 145-147.
- ECMWF, 2006a, Metview User Guide. [www] <http://www.ecmwf.int/publications/manuals/metview/index.html>
- ECMWF, 2006b, Using Customised NEdit as the Alternative Editor. [www] http://www.ecmwf.int/publications/manuals/metview/manual/Using_Customised_NEdit_as_the_Alternative_Editor.html
- Karhila, V., 2003. Metview developments. In: Proceedings of the 9th Workshop on Meteorological Operational Systems, Reading, England, November 10-14, pp.119-120.
- Lami-Thépaut, S., 2003. MAGICS next generation: an object-oriented architecture with a new contouring package. In: Proceedings of the 9th Workshop on Meteorological Operational Systems, Reading, England, November 10-14, pp.117-118.