

Meeting the challenges of the next generation of user interfaces

Iain Russell, Sylvie Lamy-Thépaut, Graphics Section, ECMWF

Abstract

Metview is a meteorological workstation designed to be a complete working environment for both the operational and research meteorologist. Its capabilities include powerful data access, processing and visualisation. It currently uses MAGICS 6 as its plotting library, but is being adapted to use the newest version, Magics++. This raises some interesting questions about the future of Metview's user interface. Metview and Magics++ are presented in this paper, along with current developments in both.

Introduction to Metview

Metview is an interactive meteorological application designed to access, manipulate and visualise meteorological data on UNIX workstations.

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

Metview was conceived as having a modular architecture, where modules can be run on difference machines. Modules can be seen as 'mini applications' which sit on top of Metview's kernel, adding both essential and optional functionality. Through this method, Metview can be easily extended by the addition of new modules.

Standard software libraries such as OpenGL, Motif and PNG are used within Metview, easing its portability within the UNIX world. Metview has been installed on UNIX systems such as IBM's AIX and various Linux distributions such as SuSE, RedHat and Ubuntu.

Using Metview

Metview can be run in either of two ways: interactively or in batch mode. The interactive user interface is an icon-driven drag-and-drop style interface. Figure 1 shows Metview's user interface in a typical working environment.

Every concept in Metview is represented by an icon. For instance, a GRIB file is displayed as an icon which can, through a right-click menu, be visualised or have its header information displayed. Plotting attributes are grouped together and called 'visual definitions'; for example, a Contour Plotting icon contains all of the parameters that can be specified when plotting contours. Such icons can be edited through an intelligent icon editor which ensures that only sensible combinations of parameters are entered.

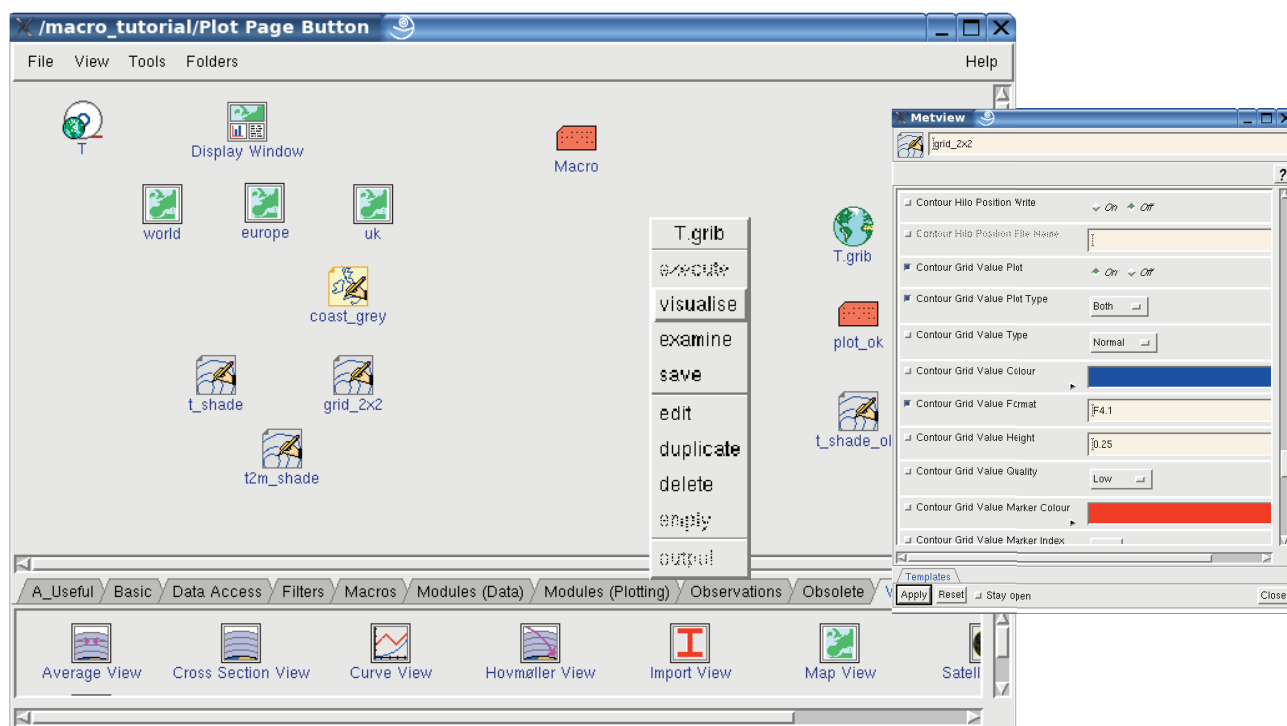


Fig. 1 The Metview working environment

Metview's Macro language provides powerful scripting capabilities to the Metview user. Designed for meteorological use, its built-in data types include fieldsets (for GRIB data) and geopoints (for scattered point data). Its extensive list of built-in functions and operators is a result of years of feedback and requests from users. The supplied Macro editor has the powerful capability of converting any Metview icon into the equivalent Macro code; a user may edit an icon and then drag it into the Macro editor which will generate the code, saving time and ensuring that the code is valid. An external editor may also be selected, and Metview comes with the means to extend the NEdit editor, giving it drop-down Macro function lists, syntax highlighting and function calltip help.

Metview can read a variety of data formats including WMO GRIB and BUFR files; data archives such as ECMWF's MARS can also be accessed.

Metview's data plotting capabilities are currently powered by MAGICs 6, ECMWF's meteorological graphics library. It can generate such varied plots as contours, wind flags, cross sections and scatter plots, organising their layout as the user requires. Work is underway to incorporate the new Magics++ as Metview's enhanced graphics library.

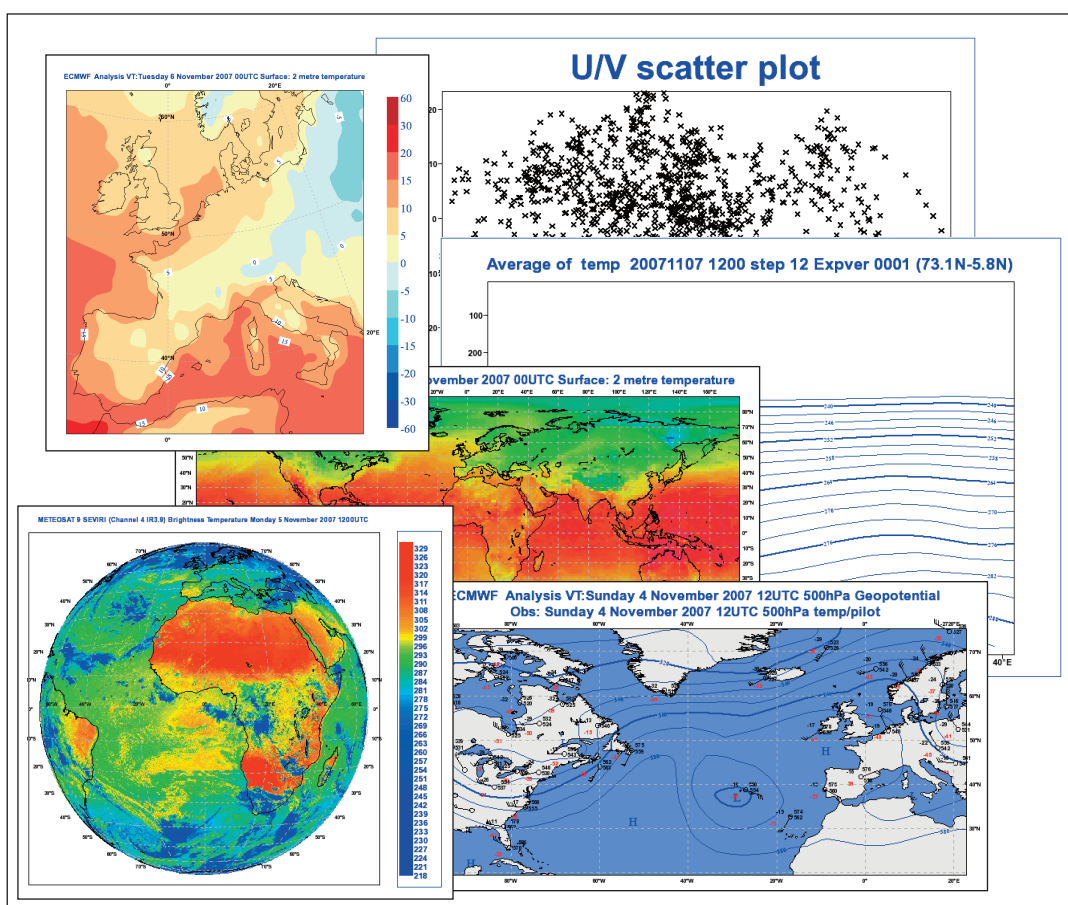


Fig. 2 A selection of plots produced by Metview

For interactive data visualisation, Metview's Display Window provides a rich set of tools which allow the user to explore their data in detail. Facilities for zooming into areas, changing the projection and producing an animation are available. A button allows the user to generate a macro program which can be used to reproduce the displayed plot automatically. Figure 3 shows the Display Window with its animation controls exposed.

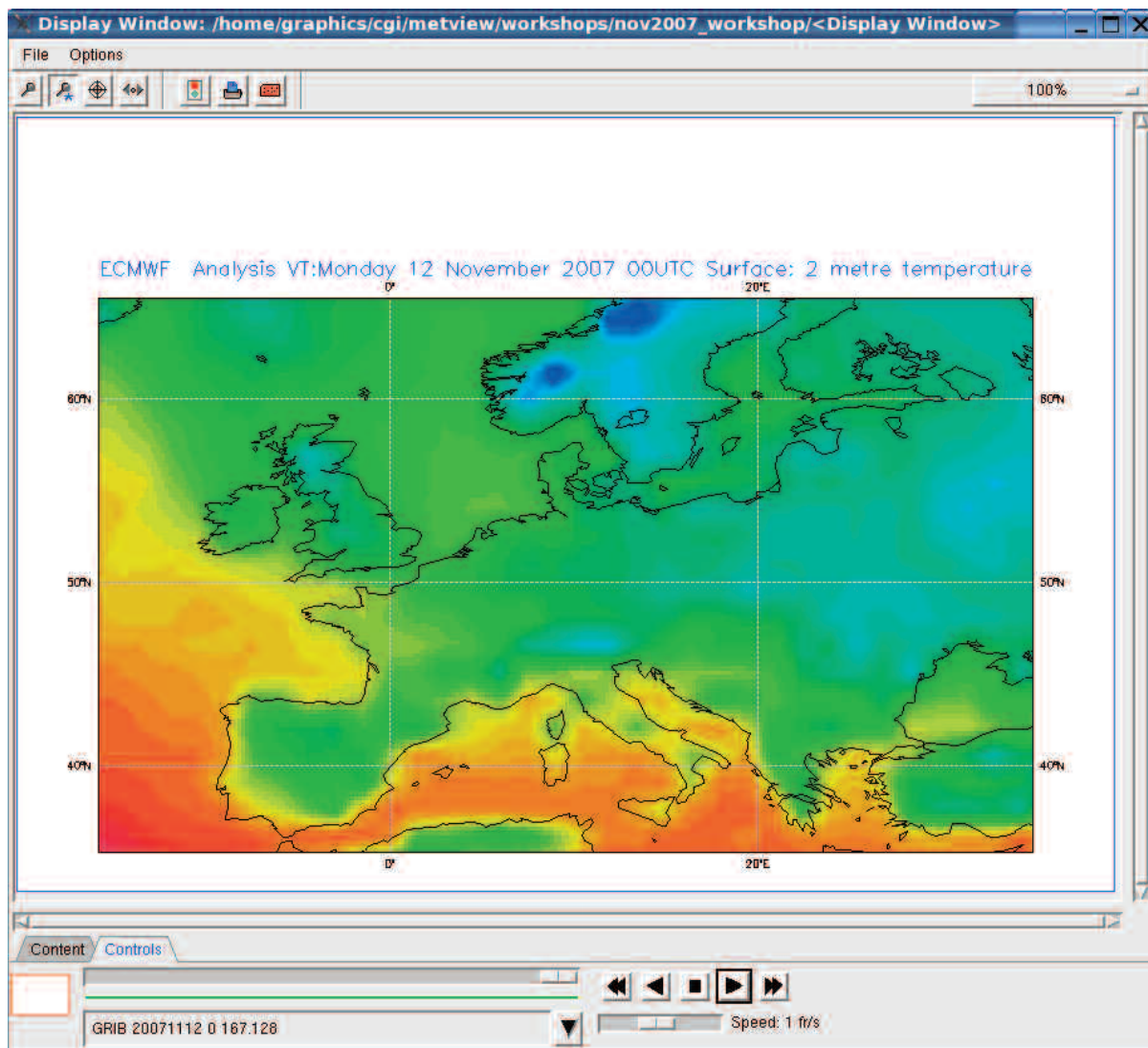


Fig. 3 Metview's Display Window with animation controls

There have been many new features and improvements in Metview since the last workshop. These include a new Percentile application, new Macro functions for creating geopot variables in memory and a set of performance timing Macro functions. Also available now is a Quick Installation Guide for those responsible for installing Metview.

In its almost 15 years of existence, Metview has been installed in more than 50 organisations around the world. It is distributed as source code, and as part of Metview's support package there is an annual training course held at ECMWF.

Current Developments in Metview

One area in which Metview should be made easier is its installation process. Investigations are in progress to determine the feasibility of moving from Metview's custom build process to the use of the GNU Autotools. These tools create a 'configure' script which is run at the installation site, creating Makefiles which are tailored to the local environment. This system is already used in Magics++, so there is some experience to take advantage of. It may turn out that this is not the way to go, but improvements will be made in one way or another.

As GRIB 2 is becoming more widely used, Metview must support it. A prototype version of Metview exists which uses ECMWF's GRIB API library instead of GRIBEX for the reading and manipulation of GRIB data. This library handles both GRIB 1 and 2 data transparently. The GRIB API library also has some point-extraction features which may allow applications such as Cross Section, which only require access to a few grid points, to be made more efficient.

A set of example Macros which users can consult is hoped to be made available in the future. These would help show the potential of Metview and provide example code which can be taken and modified. Also, a set of user-callable functions written in Macro would help simplify user code by storing useful Macro code centrally.

Work on a new plotting engine using Magics++ is in progress. The first step is to convert the input. The plotting parameters, both from icons and Macros, are stored in data structures internally called definitions. A subset of these can currently be converted into their equivalent Magics++ objects and processed for display. The new Magics++ OpenGL driver then plots the output in a new Display Window. This Display Window is a modified version of the current one, and allows some interaction with the graphical elements; for example, the user can move the plot title around the page by dragging it. However, in order to fully realise the potential of Magics++, a completely new Display Window should be developed.

Future Focus for Metview

The major focus of future work on Metview will be full integration with Magics++. This will involve developing a completely new Display Window which will take full advantage of Magics++. It should have a more intuitive concept of layers, for instance, and allow the user to select and modify elements directly in the plot. Once MAGICS 6 is no longer a part of Metview, it should be possible to build a completely 64-bit version, allowing access to far greater amounts of memory.

The future of Metview's user interface is open to question. With Motif development and support slowing down, perhaps it is time to consider other options for revitalising Metview; for instance, GTK and QT offer possibilities for a more modern feel and improved functionality. Recent developments in Magics++ have been putting more interactivity into outputs suitable for the web (SVG, PNG + JavaScript). When considered in the light of new technologies such as Adobe AIR, Mozilla PRISM and Microsoft Silverlight which are taking web applications to the desktop, this raises the question of whether the time is right to consider a web-based interface for Metview.

Meteorology is an area where large data sets are used, and heavy computations performed. Thus it makes sense to use local resources rather than transfer all this data across the web. Therefore, a web-based Metview user interface would run on a local web server (although it could also be run over a network when data volumes are not an issue). Various user interface libraries for web applications are reaching the point of offering desktop-like functionality in a web browser; JavaScript libraries such as jQuery and YUI (the Yahoo! User Interface Library) demonstrate what can be done.

Questions remain about the debugging tools for these technologies; C/C++ debugging tools have been developing for decades, but the equivalents for web tools have not yet reached the same level of maturity. Likewise, the maintainability of JavaScript and the associated web server software is unlikely to equal that of C/C++ code. These factors would make a web-based user interface more challenging from the developers' point of view.

Interactive Magics++

The future of Metview and of the interactive meteorological workstations made us design and develop a completely new aspect to the Magics software: the interactive aspect. Magics++ is still meteorologically-oriented, but now offers interactive functions that enable a user application to create a powerful tool tailored to the needs of its users (researchers, forecasters, or web clients).

These interactive aspects have been designed in parallel in two different environments: the first one is desktop-oriented using an OpenGL driver and the X/Motif toolkit. The second is web-oriented using a gif driver, the JavaScript-on-demand technology and the jQuery library. Working in these two environments allows us to be sure that the software will behave consistently in both environments.

The list of desired functionalities has been established from 15 years of Metview user feedback, and from our web users.

On top, we find the ability to navigate the maps and to select some points or areas. After the selection, the callback of the user application is executed, enabling the application to take further actions (such as zooming, or requesting a metgram).

A *layer* concept has been designed: a plot is a combination of several application-defined layers. Each layer can be toggled on or off, and some graphical properties can be changed on the fly, but can all these interactive changes be saved? The question is still open.

At this stage of development, a user in a desktop environment using an application such as Metview, or a web-user using a web interface can define the same Magics Scene Graph and obtain the same plot. The application can then enter an interactive mode such as area selection. It will be called every time the user selects an area with the description of the selected area in user coordinates, allowing it to take further actions. As mentioned before, the new Metview visualisation module and the plot-on-demand project will benefit a lot from these new developments.

Current Status of Magics++

The development of Magics++ is still rapid. In the last few months we have worked on:

- new input data types : GRIB 2, NetCDF, ODB
- new projections: Mercator
- new visualisations: Simple PolyLine, BoxPlot
- new output formats: SVG, KML for Google Earth (see Figure 4).

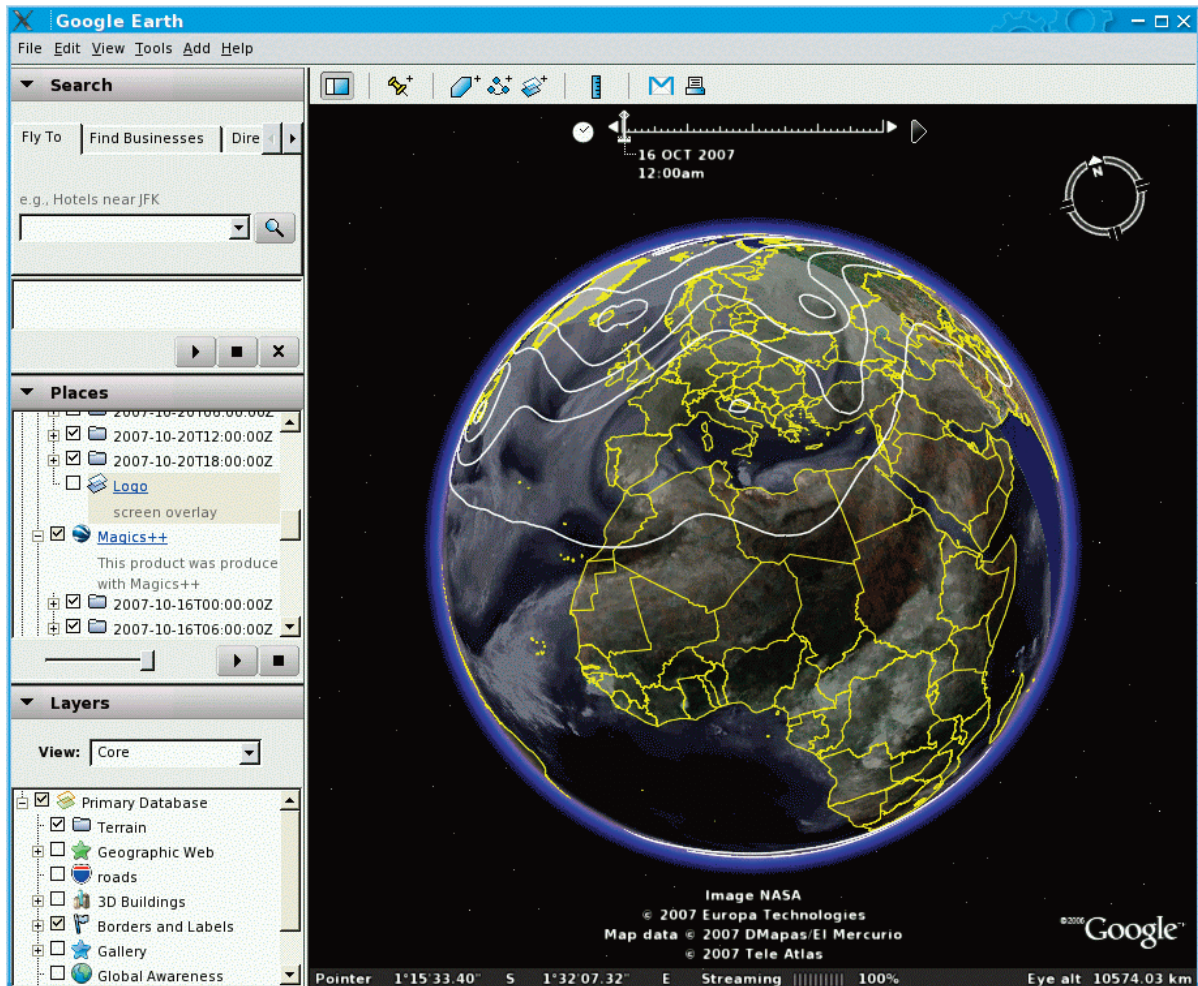


Fig. 4 Google Earth displaying output from Magics++

We have also redesigned our MagML format. MagML is an XML-based format to describe a Magics++ plot. Its purpose is to give the user an easy way to describe a plot. This description can then be interpreted by a MagML interpreter that will build the Magics Scene Graph and produce the plot. We added the notion of *definitions* and *variables* to make the use of MagML more flexible. In the latest version, MagML 3.0, the description of the visual attributes is very similar to the Metview icons, in order to provide consistency between the products of our graphics family.

```

<magics version='3.0'>
  <drivers>
    <ps name='${name=myname}' />
  </drivers>
  <definition>
    <contour id='tempe' .../>
  </definition>
  <page>
    <nopageid/>
    <map >
      <cylindrical upper_right_longitude='60'
        upper_right_latitude='60'
        lower_left_longitude='-20'
        lower_left_latitude='20' />
      <plot>
        <grib input_file_name = '${grib=t850.grb}' />
        <contour use_id='tempe' />
      </plot>
      <coastlines/>
    </map>
  </page>
</magics>

```

To interpret:

```
magmlx template.magml -grib=today.grib -name=today
```

Magics++ 2.4 ready for export

The latest version is now ready for export. A lot of work has been done to clean the previous FORTRAN interface and to improve the quality of the outputs.

The Apache license makes Magics++ available freely to everyone.