mean route method performs best. In cases where this method is most likely to fail (e.g. when there is a bifucation) one could simply choose another method.

The ship routing described in this paper is a research project. It was set up to show whether, and how, ship routing could benefit from ensemble prediction of wind and waves. There are some prominent differences to operational ship routing because the rotational motions of the ship and their effect on the ship and the cargo are not included in the cost function. Although this may not be so relevant for the statistical considerations in this article, it can be serious for a real ship in specific conditions. Also, the forecast was not updated during the 6½-day crossing. Although this was useful in this ship routing project to enhance uncertainties of weather prediction, in today's operational ship routing business, with proper means of satellite communication, updates of the current optimal ship route are done regularly.

Peter Janssen

METVIEW
Meteorological data visualisation and processing software at ECMWF

Metview is the ECMWF's meteorological data visualisation and processing tool. By virtue of its design and extensive range of features, Metview can act as a complete working environment for the operational and research meteorologist by providing powerful data management visualisation and processing tools. It can be seen as a meteorological desktop plotting package, thanks to its WYSIWYG visualisation, but it is also a powerful meteorological data-processing software thanks to its macro language, and can be used for routine production of meteorological charts in an operational environment.

Metview is based on ECMWF software for data access (MARS) and graphics (MAGICS). These reliable "workhorse" applications were in routine use at the time of Metview development and provide its underlying data retrieval/handling and plotting capability.

Metview's user interface is based on MOTIF and the X Windows system. Metview was designed for the UNIX environment and it is highly portable within the UNIX world. At ECMWF, Metview is used on SGI systems, but elsewhere it has been installed and performs operational work also on SUN, HP and DEC workstations. It has recently been ported to LINUX/GNU, thus widening considerably its potential for deployment by being able to run on comparatively very low-cost hardware. Metview has a modular architecture and was conceived as a fully distributed system, its modules being able to run on different machines

Metview was developed as part of a co-operative project between ECMWF and INPE/CPTEC, Brazil with assistance from Météo-France. The development of Metview started in 1991 and the first release became available to internal ECMWF users in December 1993. Metview was released to ECMWF Member States in October 1995. It is now used in most Members States, plus Brazil and Australia.

Since this first release, Metview has gone through a number of operational versions, all of which were structurally similar. Metview has now undergone its first major structural revision embodied in the new version about to be released. This article presents Metview to a first time user readership. Readers already familiar with Metview will note the modifications it has undergone.

Metview's new version presents a revised main User Interface with improved functionality where user customisation was a main priority. The greatest changes are in the visualisation tool, which was completely rewritten from scratch according to a different design principle. Internally extensive rewriting, streamlining and simplification of the code took place, which will make Metview easier to extend and maintain.

The Metview icon Concept

Metview can be described as a tool working on archived data (model fields, observations, user data) to produce either:
- a visualisation of the data or of a derived parameter (on screen, on file, on paper) - this can vary in complexity and format (from a contoured geographical field to an x-y plot;

and/or
- a file on disk containing the retrieved data or some derived data - the file can be binary or ASCII and its content can vary in complexity.

The process from original data to final output constitutes a Metview task. This can range widely in complexity but the important concept is that:
- any Metview task can be decomposed into separate steps arranged in a logical sequence;
- each step is defined (specified) by the user - these definitions are represented in the GUI by Metview icons. The user operates on these icons to carry out each step.

Metview icons (definitions) are the components of Metview (from the user point of view). In fact, the design principle (paradigm) of Metview can be summarised by:

Everything in Metview is an icon,

while the operating principle of Metview is:

Every Metview task is a sequence of actions on icons.

This article shows how these principles are realised in practice.

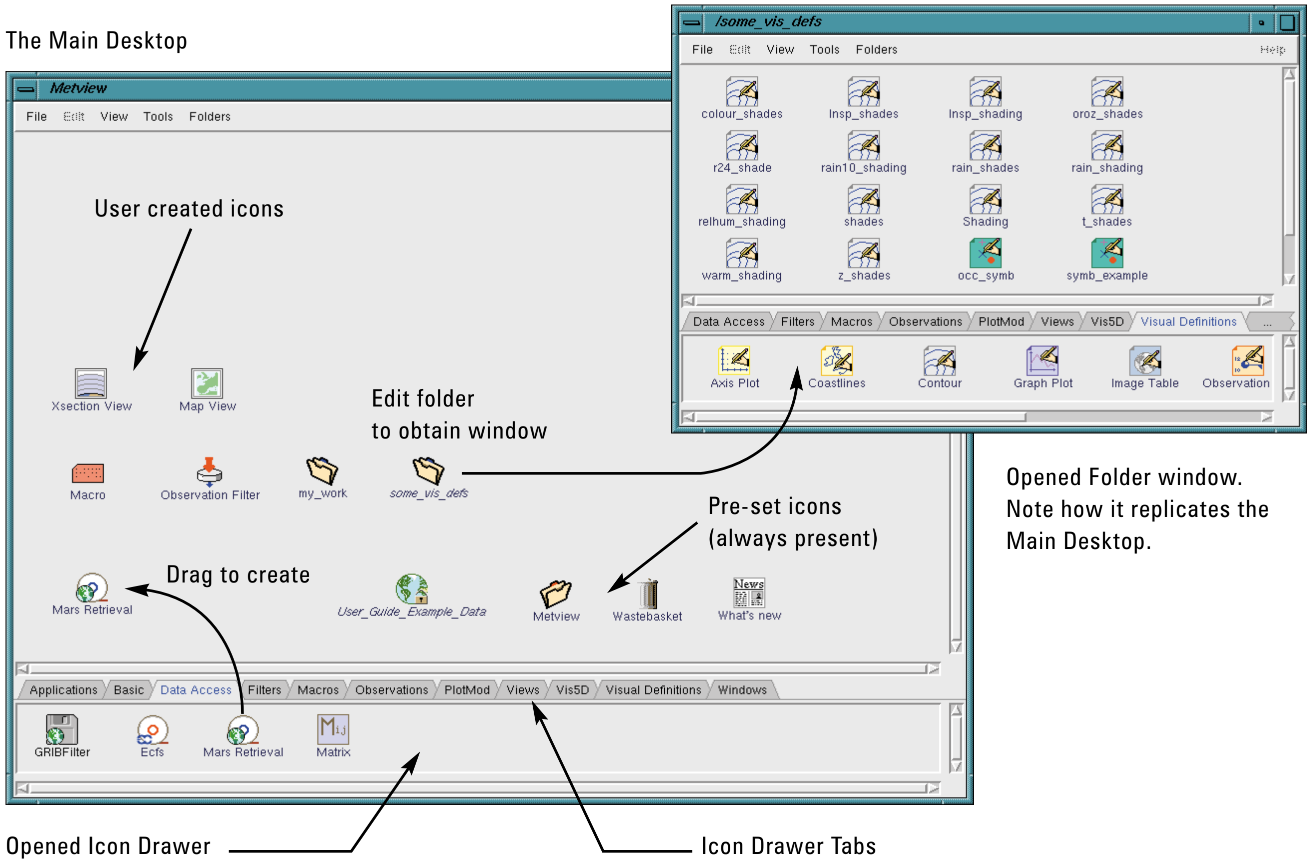

Figure 1. The Metview User Interface or Desktop. The four icons on the lower right part of the main window are predefined and always appear on start up. Folder icons open to reveal a new (replicated) desktop. Folders can be symbolic links, in which case their name appears in italics. Some examples of user created icons can be seen in the desktop. Every user task is carried out by means of operations (actions) on suitable icons. Icon templates are kept in icon drawers from where they can be dragged to create new instances on the user desktop.

The Metview Environment

Users launch Metview from a UNIX command line with the following command:

% metview

The first thing a user sees is the Metview Desktop as shown in Figure 1. Within this desktop you have icons and these can represent actual data, data requests, processing or plotting specifications, macro programs and applications.

Metview aims to create a complete interactive working environment. You can conceivably work exclusively on the Metview desktop. Metview icons can represent your working elements, from scraps of text to GRIB files, database requests, data filters, plotting specifications, etc. You can run UNIX shell scripts and Metview includes its own Mail tool with a facility for seamless sending of icons, so you can communicate with colleagues (e.g. exchange data and macros) and developers (e.g. questions, bug reporting).

What you see as the main desktop is in fact a UNIX directory, called **metview**, residing on your root directory. Every folder within the main user desktop is a sub-directory of this **metview** directory.

Given that symbolic links within the metview directory are also recognised as folders in the main user desktop, you can have your scratch space, or some data storage directory, present on the desktop. Its contents (for instance GRIB files) are transparently accessible to the user's visualisation and processing work. Should suitable permissions be in place, you can access other user's directories within the Metview desktop, thus facilitating exchange of work and ideas between users.

Metview can handle several data/file types:

- GRIB for fields and satellite images
- BUFR for observations
- ASCII matrices of lat-long gridded data
- Geopoints (sets of irregularly spaced data points in ASCII)
- NetCDF for data other than model fields and point data (e.g. cross sections)

Users create most of the icons themselves. However, if a file is copied or brought in to the metview directory (or one of its subdirectories) it is automatically assigned an icon. There are a few file types which are recognised and automatically assigned a specific icon – those that Metview can handle (see above) plus shell scripts, PostScript files and plain text files; otherwise a generic binary file icon is used.

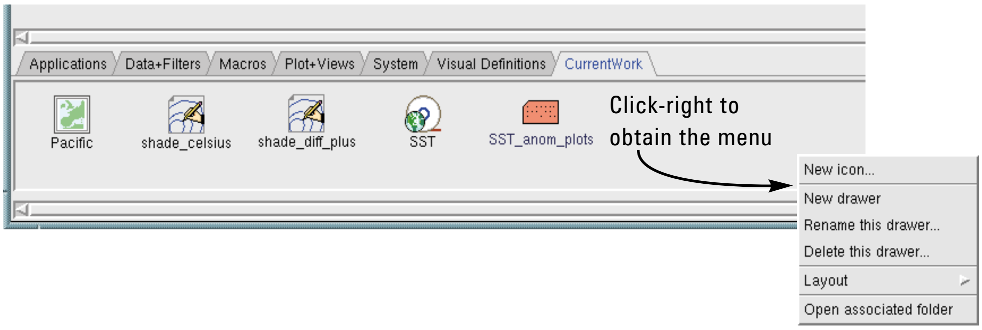

Figure 2: Example of a user-customised group of icon drawers. Note the renaming of most drawers, deletion of others the user will not use, and the creation of a drawer for a user project containing icons pre-specified with required parameters. The icon drawer menu shown on the right provides most of the functionality for these changes.

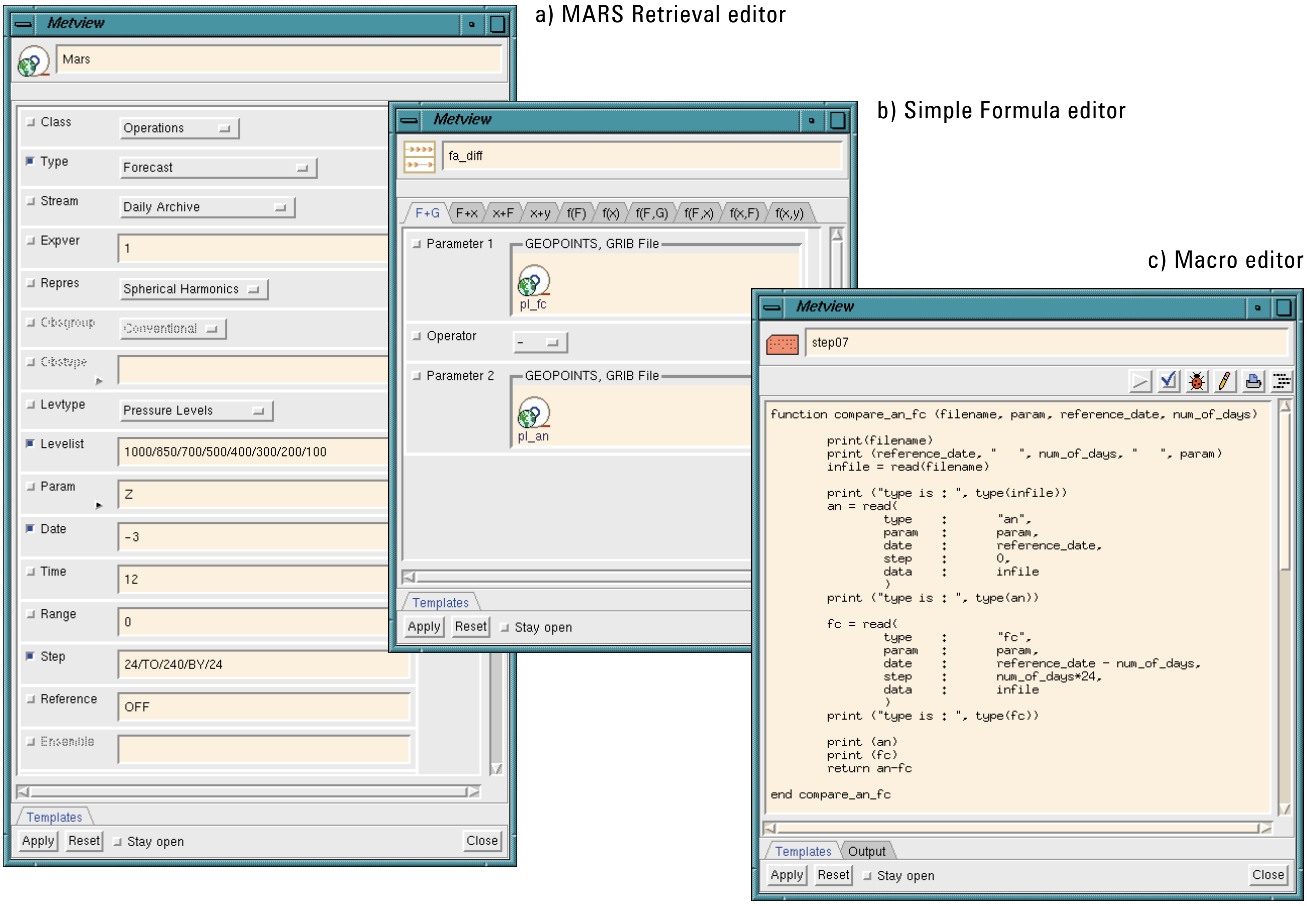

Figure 3: Examples of icon editors – (a) a standard editor (MARS Retrieval) with parameters set by means of text boxes and option menus, (b) a family editor (Simple Formula) with different interfaces in the same editor each selected by its own tab and (c) a macro editor, a plain text editor with extra functionality for code development (top row buttons).

Working with Metview Icons

The fundamental unit of Metview is the icon. Here we see how users can create icons and how one operates on icons to achieve a given outcome.

As you can see in Figure 1, icons are arranged in icon drawers. Each drawer reflects the icon types it contains, so you find drawers for visual definitions, for data icons, for macro-like operators, etc. You have full control over the arrangement of the drawers on your desktop and you can:

- name drawers;
- create new drawers;
- delete existing drawers.

The drawer contents are also subject to customisation. You can:

- remove or rename icons;
- transfer icons from drawer to drawer;
- transfer user created icons from the desktop to a drawer.

So you can group icons according to your own convenience and even create customised icons and group them in drawers to act as user icon libraries – e.g. you may have a drawer for specific types of data retrievals, or geographical areas, or plot window layouts (see Fig. 2)

Icons can be created by:
- dragging from an icon drawer to the main or folder desktop (as shown in Figure 1);
- clicking-right on a desktop or a drawer – this launches an icon list from where the required icon can be created by clicking on it;
- duplicating an existing (similar) icon of the same type and modifying it accordingly.

Once the created icon appears on the desktop, it must be edited to conform to your requirements. The icon editor window allows you to enter values for parameters by means of text entries, option menus, toggle buttons, icon drop fields and colour editors. Once you save and exit, the icon is then ready to be operated upon.

All Metview (interactive) work is carried out through actions (operations) on icons. To operate on an icon you click right on the icon and choose one from:

Execute Visualise Examine Save Drop Edit Duplicate Delete

The type of icon determines which actions apply to it. The last three actions in the list above apply to all icons and are self-explanatory. The first five are not applicable to all icons – e.g. you can't *Visualise* a visual definition icon since this type of icon only specifies how to contour data.

Visualisation in Metview

The visualisation concept in Metview has changed considerably from the current version to the new version. Previous functionality has been kept (e.g. point value query, re-projection and re-sizing,...) while new capabilities were added. The most striking new capability is the ability to display different types of plots in any arbitrary layout featuring effective WYSIWYG. Users may now display concurrently:
- 2-dimensional geographical plots – meteorological fields, observations, station locations, satellite images;
- cross-sections along defined spatial limits;
- tephigrams;
- vertical profiles of parameters for points or areas;
- time series of parameters for a given point;
- curves derived from users' data.

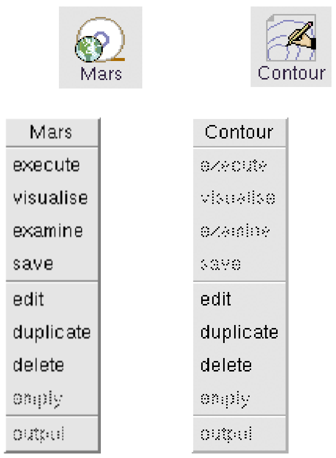

Figure 4: Examples of icon menus. The menu presents a choice of action to the user. Some actions may not be available as you can see from the menu on the contour icon.

Throughout, we refer to *visualisation* in a broad sense to include:
- on-screen display with interactive facilities;
- output to file, in PS (for printing) and PNG formats (Web publishing, presentation);
- hard-copy print-out (paper/transparency).

Central to the new visualisation module is the concept of a (plotting) view. Essentially, the view specifies how to plot a given data. For example, from a data unit composed of multi-level sets of fields for a number of forecast steps, a user can produce a set of 2D geographic fields, a time series of cross sections or vertical profiles or a set of time series plots for each level. Users have a view for each of the plots they may want to prepare from the data. Currently Metview provides:
- **Map View** (for 2D geographic plots) the user specifies area and projection;
- **Cross-Section View** (for cross-section plots) user specifies the geographic co-ordinates of the extremities of a line;
- **Vertical Profile View** (for vertical profiles) user specifies the geographic co-ordinates of a point or small area;
- **Curve View** (for plotting any x-y sequence of points, from trajectories to time series).

Icons for layout and plot control

Icons for retrieval and computations

Icons for plotting specification

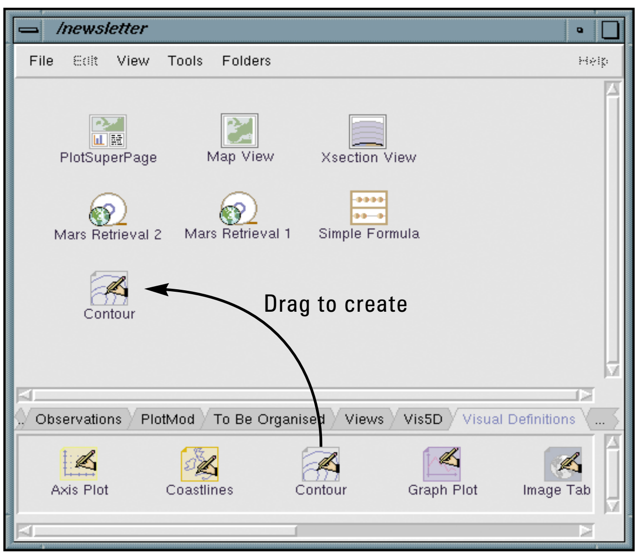

Figure 5: Folder storing icons involved in a project, in this case a plain forecast-analysis difference. The icons present have just been dragged from their drawers. They will now be edited and modified according to requirements.

The principle behind the Metview visualisation tool is that the user specifies how he/she wants the data to be plotted by means of a view and is able to mix different views in the same display/paper sheet. The data unit passed to the display window will generate the appropriate plots, cascading through the different plotting perspectives. Users can still visualise data icons directly, without specifying anything relative to layout or view, in which case suitable defaults apply (e.g. single map plot on A4 landscape).

To clarify how Metview works, let us go through a schematic visualisation - visualising (forecast-analysis) differences both as a map over a given region and as a cross-section along a given transect. We'll create a folder for this task and create the icons inside it by dragging from their drawers (see Figure 5). Having assembled the icons, they have to be edited to conform to the user's requirements

To derive an forecast-analysis difference, a user has to retrieve forecast and analysis fields (at multiple levels) using the **MARS Retrieval** icons and compute their difference with the **Simple Formula** icon. Preparation of the forecast data icon is exemplified in Figure 6 as a sequence of edit-modify-save actions. Once the icon is prepared it can be used, in this case as input for a **Simple Formula** icon.

The **MARS Retrieval** icon is used for retrieval of any data present in the archives. Other institutions will have to adapt to their own archive/database retrieval procedures during Metview installation, but the principle will remain the same. Here we are using the **MARS Retrieval** icon as input to another icon, but you could have visualised its specified data directly or you could have used the **Save** action to obtain a GRIB file with that data.

The **Simple Formula** icon is a general icon for simple calculations with fieldsets, point values and scalars. You can specify an operation between fields, fields and point values, fields and scalar values, functions of scalars and functions of fieldsets. The icon can be used as input to yet another icon, visualised or saved to derive a GRIB or geopoints file with the result of the calculations. The **Simple Formula** icon illustrates an important Metview feature - many icons take data icons as input. These *embedded* icons are automatically activated upon operating on the external icon.

Similarly, the **View** icons have to edited and modified from the original defaults to the requirements. In this case we specify a cylindrical projection European window from (70°N, 20°W) to (20°N, 40°E) (top left, lower right) for the **Map View** and a line from (20°N, 20°W) to (70°N, 40°E) for the **Cross Section View**.

In order to define how the final plot will look, the visualisation window is specified with a **PlotSuperPage** icon. Edit this icon to reveal its graphical interface (see Figure 7). The editor allows users to define the size and orientation of the plot surface and divide the plot surface into any arrangement of arbitrarily sized fractions. In this example, we'll define an A4 landscape plot divided vertically in two equal sized plots. The result is saved under the name **map_xs**.

Contour icons specify how you want the plots to be contoured. The editor provides for a wide-ranging choice of types, using lines, shades and patterns. Since we are plotting differences we specify a contour such that very small differences are not plotted, positive differences (forecast warmer than analysis) are plotted in warm tones and negative differences (forecast colder than analysis) are plotted in cold tones.

Metview provides visual definitions other than contouring. Users have available definitions for meteorological observation plotting, symbol plotting, wind arrow/barb plotting, coastlines, plot axis, x-y curves and satellite images.

Once the **Plot Page** icon is prepared, you can carry out your visualisation. Following the example above, you would execute or visualise the plot page icon. Once the plot surface is up you can drop the **Simple Formula** icon inside each panel, followed, or in conjunction with, the contour icons.

The result is shown in Figure 8 and shows how you can visualise the same data concurrently in different perspectives. You can lay out your visualisation the way you want it and mix any views you require.

Each plot division can function independently of each other using its own data, views and contours. The plot window is equipped with features to ease your on-screen analysis. These are detailed in the next section and include:

- zoom and magnification;
- on screen manipulation of the visualisation via direct access to component icons;
- simultaneous animation of different plots;
- browsing through a stack of fields;
- re-projection and windowing of visualised data;
- querying of point values;
- printing of visualisation (WISIWYG).

Visualisation features

Once you have a visualisation up and running, you can use the features offered by the visualisation tool to help in your analysis. These are exemplified below, using different visualisations:

Zoom and magnification

These tools are available from action buttons on the plot window button. They lead to different results, in that a magnification is a purely graphic process while a zoom leads to re-calculation of the plotting (see Figure 9).

Querying point values

To check the value of fields at any location, or the values of plotted observations, the plot window provides a point tool. Click on its button, then anywhere on a field or on an observation location to produce, respectively, the interpolated field value at that location or the values of the observation parameters printed to a floating window, as shown in Figure 10.

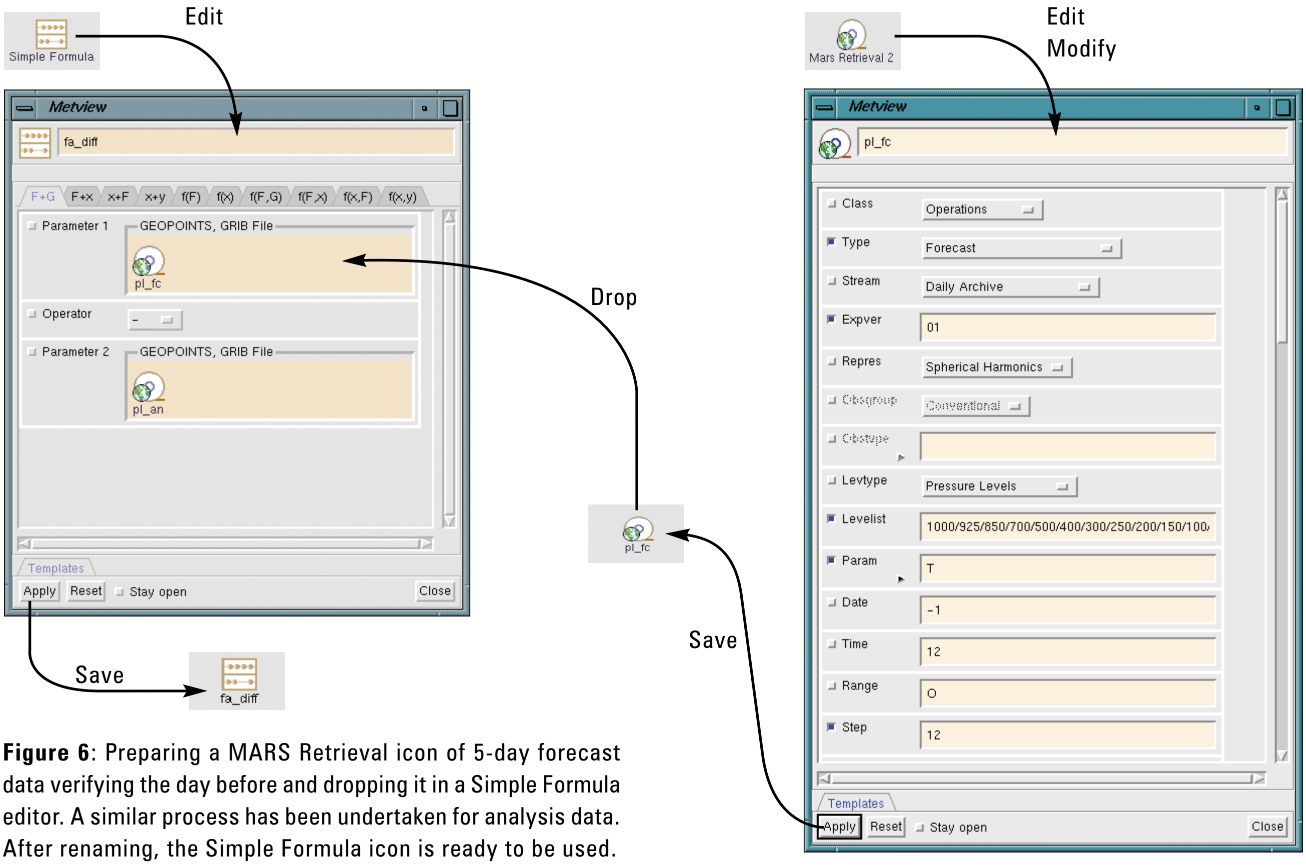

Figure 6: Preparing a MARS Retrieval icon of 5-day forecast data verifying the day before and dropping it in a Simple Formula editor. A similar process has been undertaken for analysis data. After renaming, the Simple Formula icon is ready to be used.

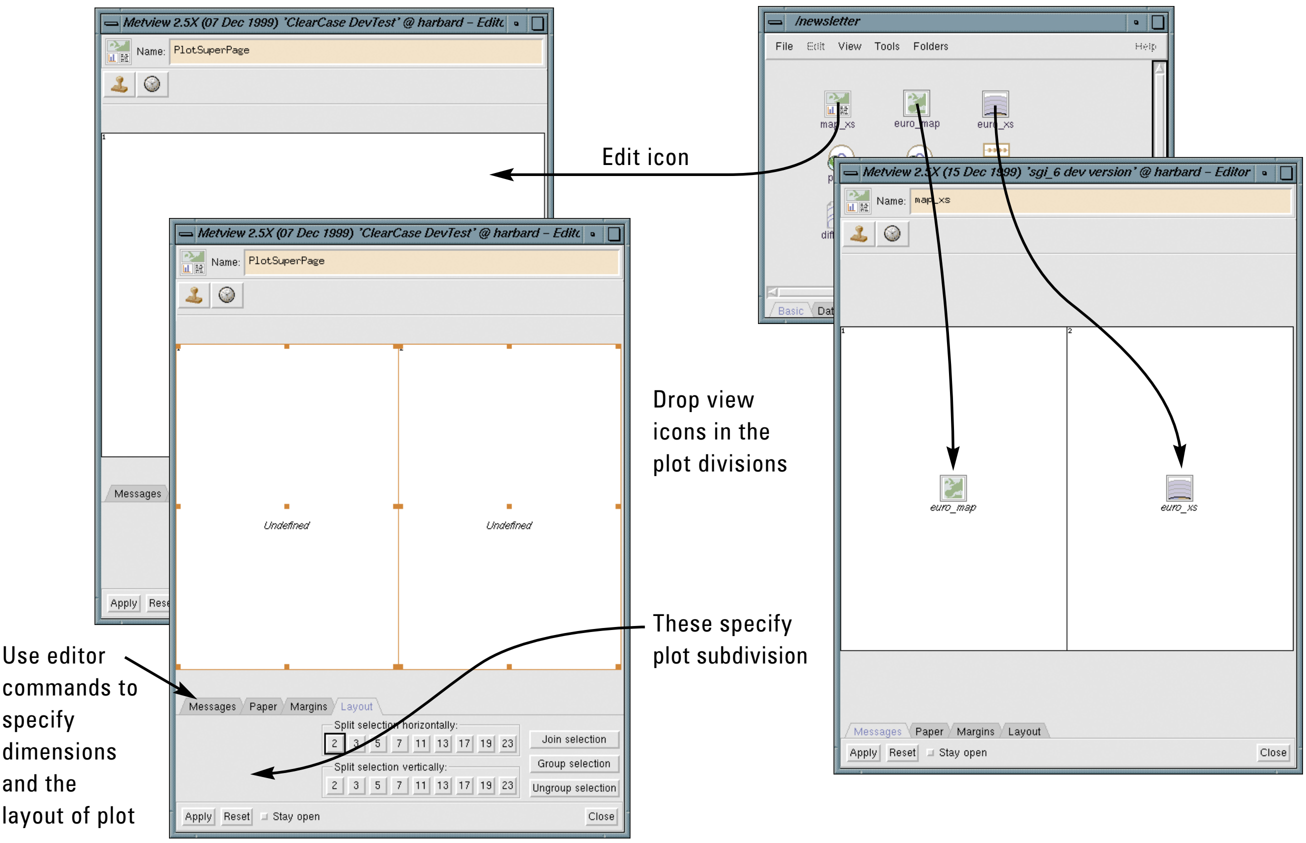

Figure 7: Interactive preparation of a visualisation layout. The initial blank plot surface is sized to A4 landscape and then subdivided in two vertically. Subdivision is achieved by selecting one or several rectangular regions and clicking on numbered buttons. Once this is ready, users can drop view icons in the subdivisions according to their needs.

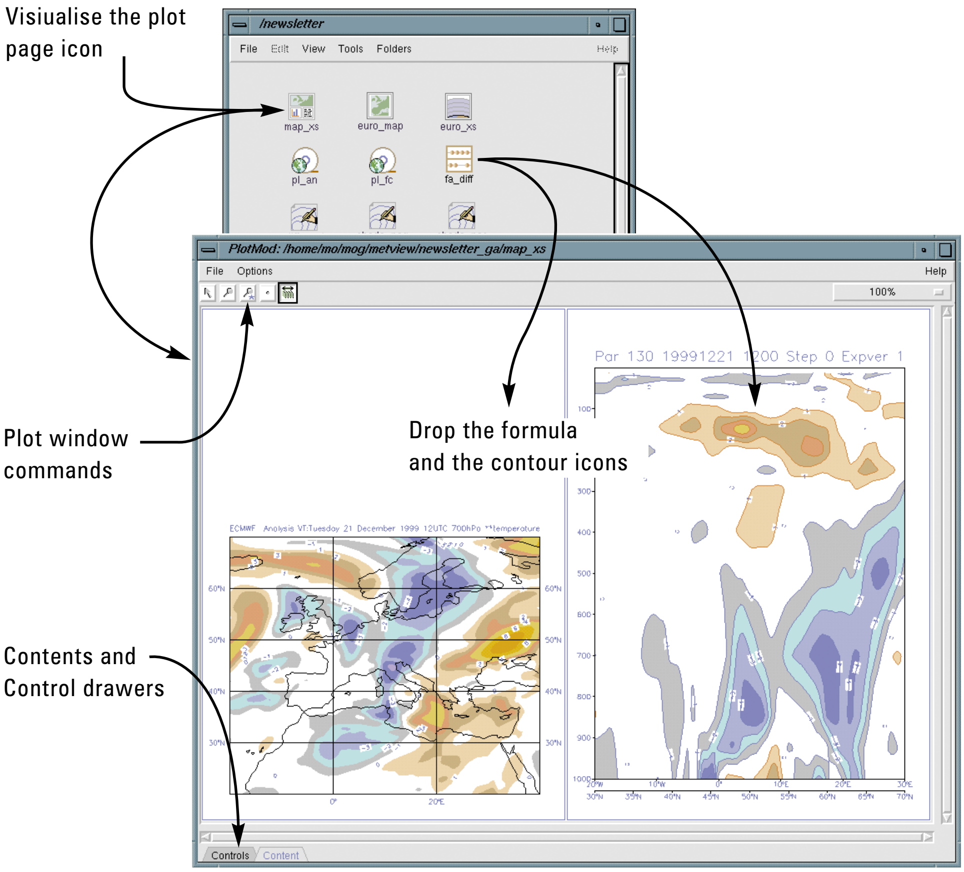

Figure 8: Result of a simple visualisation: forecast minus analysis differences as geographical fields (left) and cross-section (right). Visualise the plot page icon and then drop the data and contour icons to derive the visualisation. The two plots can have independent contours and title styles.

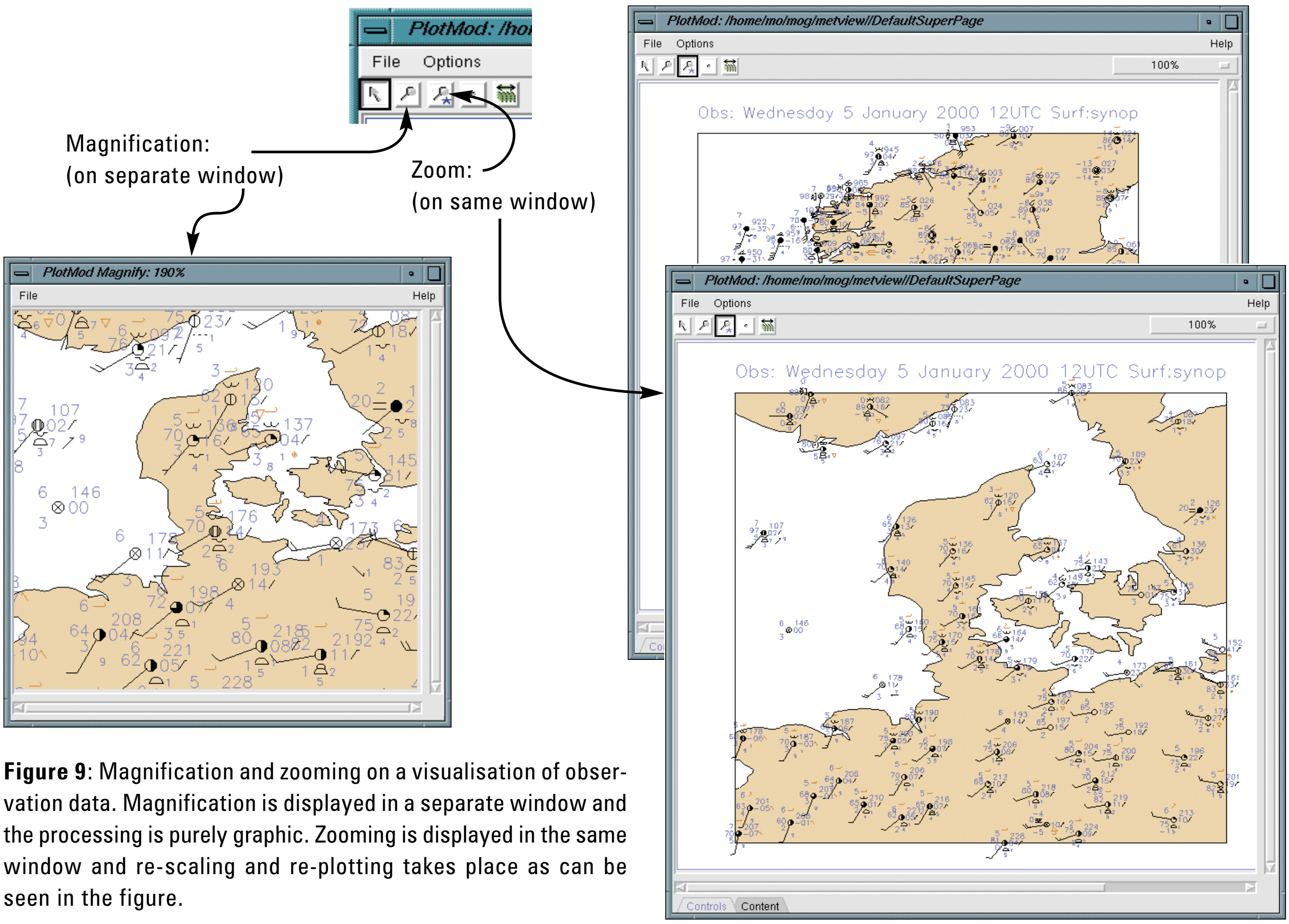

Figure 9: Magnification and zooming on a visualisation of observation data. Magnification is displayed in a separate window and the processing is purely graphic. Zooming is displayed in the same window and re-scaling and re-plotting takes place as can be seen in the figure.

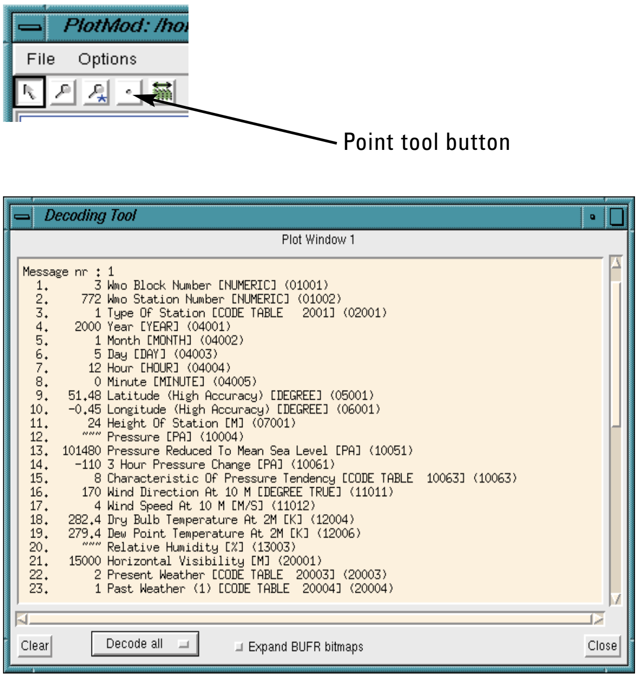

Figure 10: Example of point tool output from a visualisation of observations.

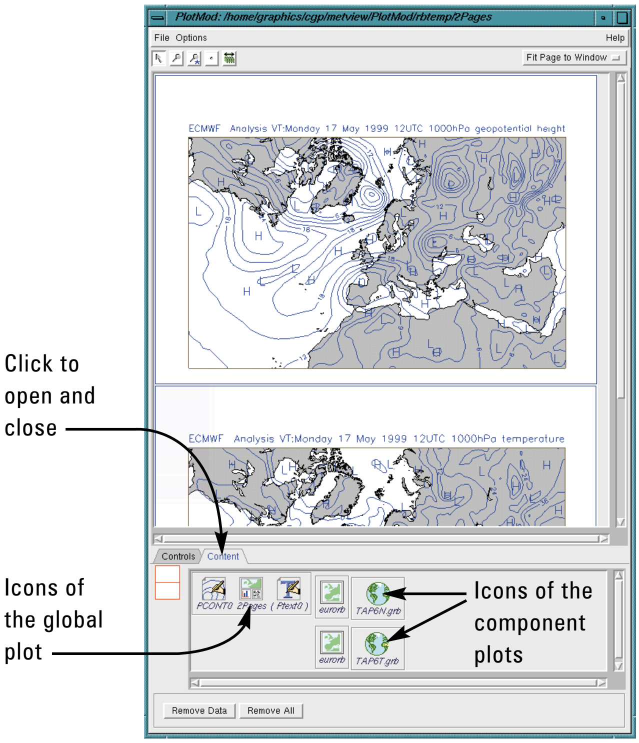

Figure 11: A sample visualisation with the contents window open. The left-most rectangle contains the icons valid over the global plot. The two areas on the right contain the icons that apply to each component plot.

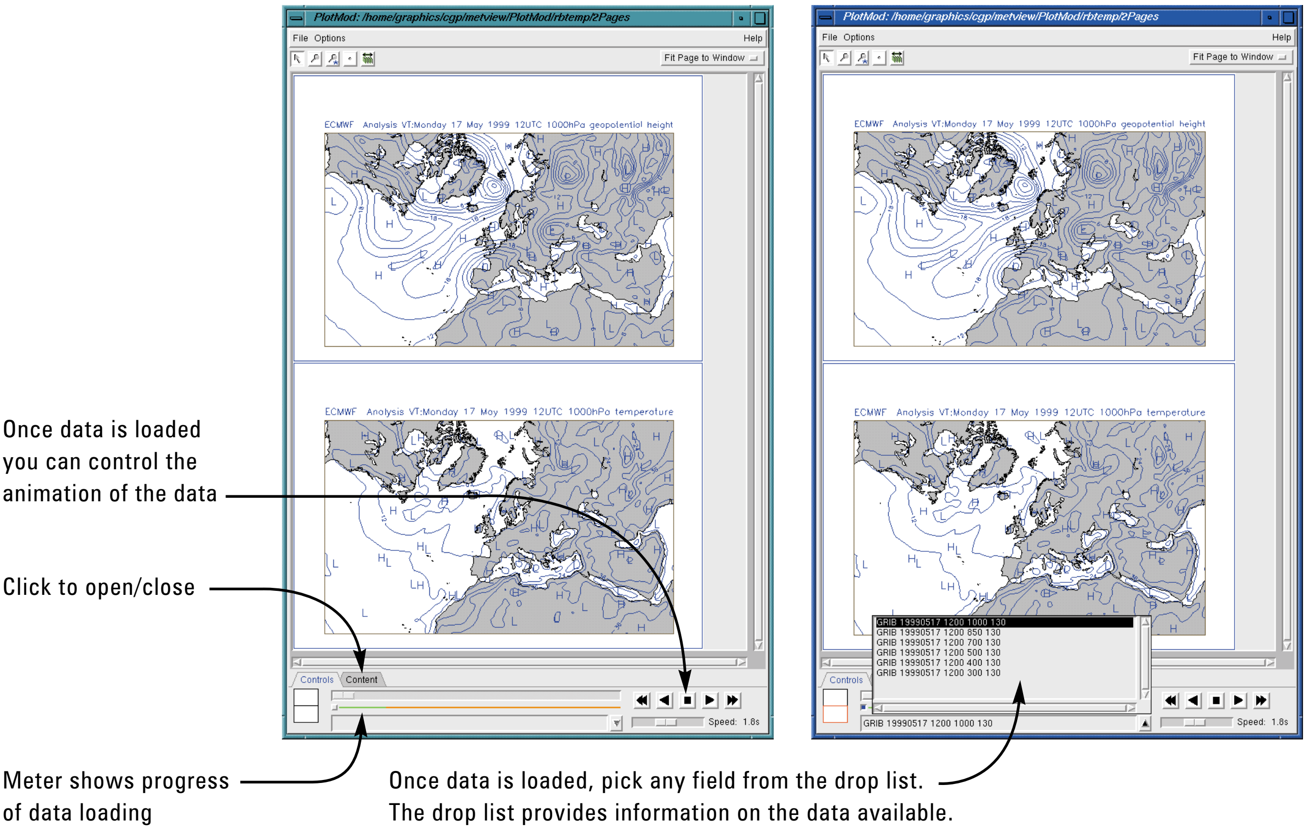

Figure 12: A sample visualisation with the controls window open to allow users to browse through a long data sequence or animate it. You can animate each plot division independently.

Visualising contents

As you have seen there are a number of icons involved in a visualisation. You require a way to manipulate each one of them individually. This is the case when you are modifying a view, tweaking a contour definition, a coastline or an axis. You may also want to add a new icon or remove an existing one. Access to visualisation components is provided through the *contents window*. As the name implies it shows you the contents of the visualisation, i.e. the icons and how they are organised in order to produce what you see on the screen. The contents window organises the component icons in rectangular areas, one for the global plot and one for each of the component plots (see Figure 11). The plot window icon is shown for reference, since it can't be modified or removed. The data icons can't be modified but can be removed. All other icons can be modified and removed. You can drop new icons in the plot rectangles - depending on which one you drop the icon in, different modifications arise on the plot - e.g. drop a data icon on the global plot contents area and the data is plotted on all component plots (cascading through the views). You can associate contour icons with a given plot or make them apply to all plots.

Animation and browsing

When you load a long sequence of fields on a visualisation, you may want to animate the sequence (e.g. a temporal sequence). You may also want to browse through the sequence, i.e. being able to pick any field in the sequence. The plot window has a *Controls* drawer which provides controls for both browsing of the data set and controls for animation (start, stop, advance, rewind). Figure 12 provides details.

Other Metview 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 extensive range of pre-defined application (icons) but also built-in utilities.

Metview provides its users with the facilities to communicate and exchange work samples with other users and development teams. This achieved by means of two built-in applications, Metview Mail to send and Metview News to receive information.

- **Metview Mail** is a simple mail tool developed to send (not receive!) messages from Metview users to other people, other Metview users, and user support and development teams. Its distinguishing feature is the ability to send icons as attachments (automatically unencoded), which can then be extracted at the destination address. It enables one user to demonstrate work results to other users. It may be used in co-operative tasks, sending bits of work back and forth between participants. Any number of icons can be mailed, and icons used as input to the mailed icon are also mailed. Mailing a folder icon will also mail all icons contained in the folder (which is a convenient and effective way to send groups of icons). A simple example is shown in Figure 13.
- **Metview News** is the application used to read Metview mail messages, provided the incoming mail has an icon in attachment. For messages without attached icons, use your UNIX mail server. Metview News is also used to install predefined examples, short tutorials and information on new features. The tool is launched from the **What's New** icon on the main desktop (one of the preset icons always present - see Figure 1).

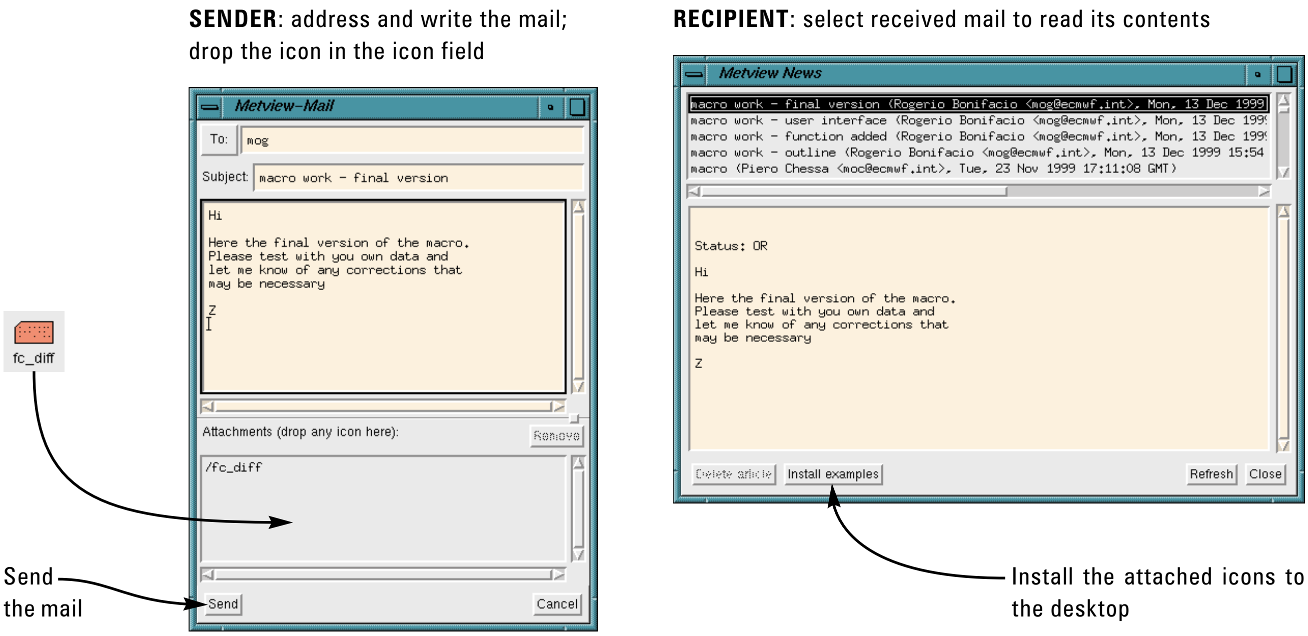

Figure 13: Working with Metview Mail and Metview News to send/receive messages and interchange icons. Your Metview administrator may also place in News some examples and short tutorials for self practice and study.

Metview provides two levels of user help:

- **A context sensitive *online help* system**: To activate, position the mouse pointer over a target (which may be an icon, a button, the toolbar, a menu choice, the desktop) and press **F1**. A window with hyperlinked information about the chosen target will appear.
- For more in-depth help, the Metview installation provides a set of *HTML manuals* that can be viewed with your system's web browser. The HTML manuals have an equivalent normal text version, provided as a set of PS files which institutions can print out to their users. The manuals are divided in three volumes, the first on general Metview usage, the second on the Metview macro language and the third an icon reference.

More direct help can be had by a built-in *Bugs Report* utility. This is simply a pre-addressed mail message, where you describe your problem and drop the troublesome icons.

Metview includes a number of meteorological applications as icons, covering a wide range of tasks. Applications are understood to be icons which can produce a new dataset from some input data. These data can be saved to a (NetCDF format) file on disk, or can be simply visualised – note that there is no difference between visualising a set of model level GRIB files in a cross-section view and visualising the result of a cross-section application on the same data. However, only the cross-section application allows you to save the cross-section data.

The Metview applications include:

- **Formula/Simple Formula** – for simple mathematical computations of GRIB fields (more complex manipulation should be done in Metview Macro);

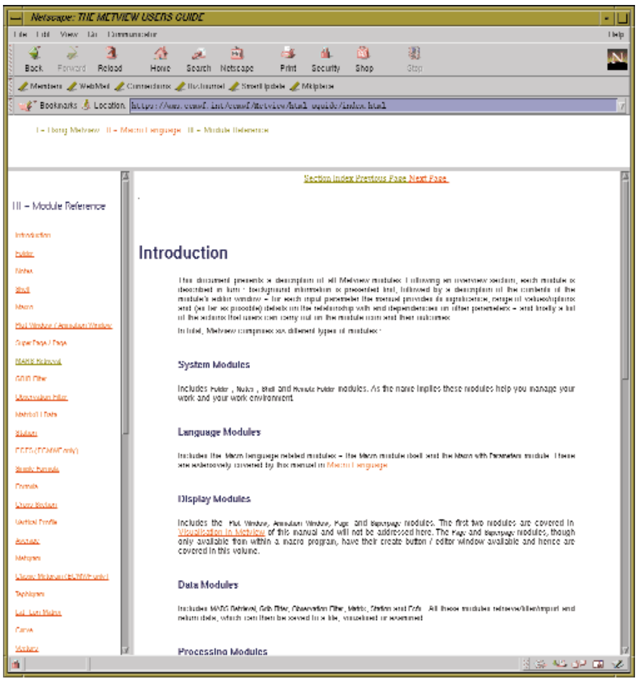

Figure 14: Metview HTML manuals – equivalent printed version exists as well.

- **Cross Section** – derives a cross section from pressure- or model-level fields;
- **Average** – derives a zonal average from pressure or model-level fields;
- **Vertical Profile** – derives a vertical profile from pressure- or model-level fields for a point or a rectangular area;
- **Tephigram** – generates tephigram plots from either model fields or observations;
- **Potential Temperature** – derives potential temperature, equivalent potential temperature and saturated equivalent potential temperature from model-level or pressure-level fields of surface pressure, temperature and humidity;
- **Vector Family** – derives vector fields from scalar fields of *u* and *v* components (Cartesian co-ordinates) or intensity and direction (polar co-ordinates). Option to colour vector arrows according to a third field (e.g. colouring wind arrows according to temperature);
- **Budget Family** – calculates and plots various budget quantities (atmospheric energy, surface energy, hydrological and top radiation);
- **Spectra** – generates plots of spectra as a function of Legendre polynomial order;
- **Velocity Potential/Streamfunction** – derives fields of velocity potential or streamfunction from input fields of divergence or vorticity;
- **Rotational/Divergent Wind** – derives fields of divergent or rotational wind vectors from input fields of divergence or vorticity;
- **Data Coverage** – generates plots and files of observations according to their quality and arrival time.

An element lacking in previous versions of Metview was the ability to produce 3D visualisations. Rather than developing such facilities from scratch, it was thought that a far more fruitful and inexpensive approach would be to integrate Metview with some other software with 3D capability. Given its features, free availability and wide user community, the strongest choice was Vis5D from the Space Science and Engineering Centre of the University of Wisconsin – Madison's Graduate School **(http://www.ssec.wisc.edu/~billh/vis5d.html)**. A co-operative venture was established between ECMWF and SSEC in order to integrate both pieces of software.

Metview was equipped with features that enable it to talk to Vis5D and Vis5D was enhanced with the ability to accept Metview icons. Vis5D is controlled via TCL/Tk scripts, so a drop of a Metview icon translates the icon's content into TCL/Tk. Metview provides the interested user with a set of Metview icons specially designed to work with Vis5D. The approach from the point of view of a Metview user is to use Vis5D as if it were a Metview application. This is to say that a Metview user can do all the data retrieval and processing within Metview and then launch Vis5D via a Metview icon.

The simple example in Figure 15 shows that you are able to control Vis5D by means of Metview icons, i.e. you can drop Metview icons that specify how Vis5D will display your data. Further data icons can be dropped in the Vis5D window. This allows you to use Vis5D without in-depth knowledge of this software thus satisfying those who only have episodic need of Vis5D abilities. On the other hand, Metview users expert with Vis5D can use Metview for retrieval and computations of data and then, once Vis5D is launched, use it as a stand-alone element. Also, since you can control Vis5D with Metview icons you can also make Vis5D run from Metview macro programs.

The Metview Macro Language

The Metview Macro language was designed to provide an easy, powerful and comprehensive way for a researcher/operator to manipulate and display meteorological data. A language is the best *user interface* to describe very complex sequences of actions, particularly if the flow of action is conditional. It 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.

It aims to be as simple as a script language (e.g. UNIX shell) to get started, and to be as powerful as a modern computer language. "To be as simple as a script language" implies that no variable declarations or program units should be required. "To be as powerful as a computer language" implies support for variables, flow control, functions and procedures, I/O and error control.

Macro programs are created with the Metview Macro editor, a text editor with added functionality such as run, debug, beautify, shown in Figure 3(c). The language is easy to learn with a relatively shallow learning curve and there are self-learning tutorials prepared for Metview Macro complementing an extensive user guide.

Metview Macro can handle (read, process and write) a variety of file types: GRIB, BUFR, Geopoints and plain ASCII data files. Input and output are done via **read()** and **write()** functions but, in keeping with the Metview Macro approach, the user does not need to specify or know details about the file type, dimensions and format.

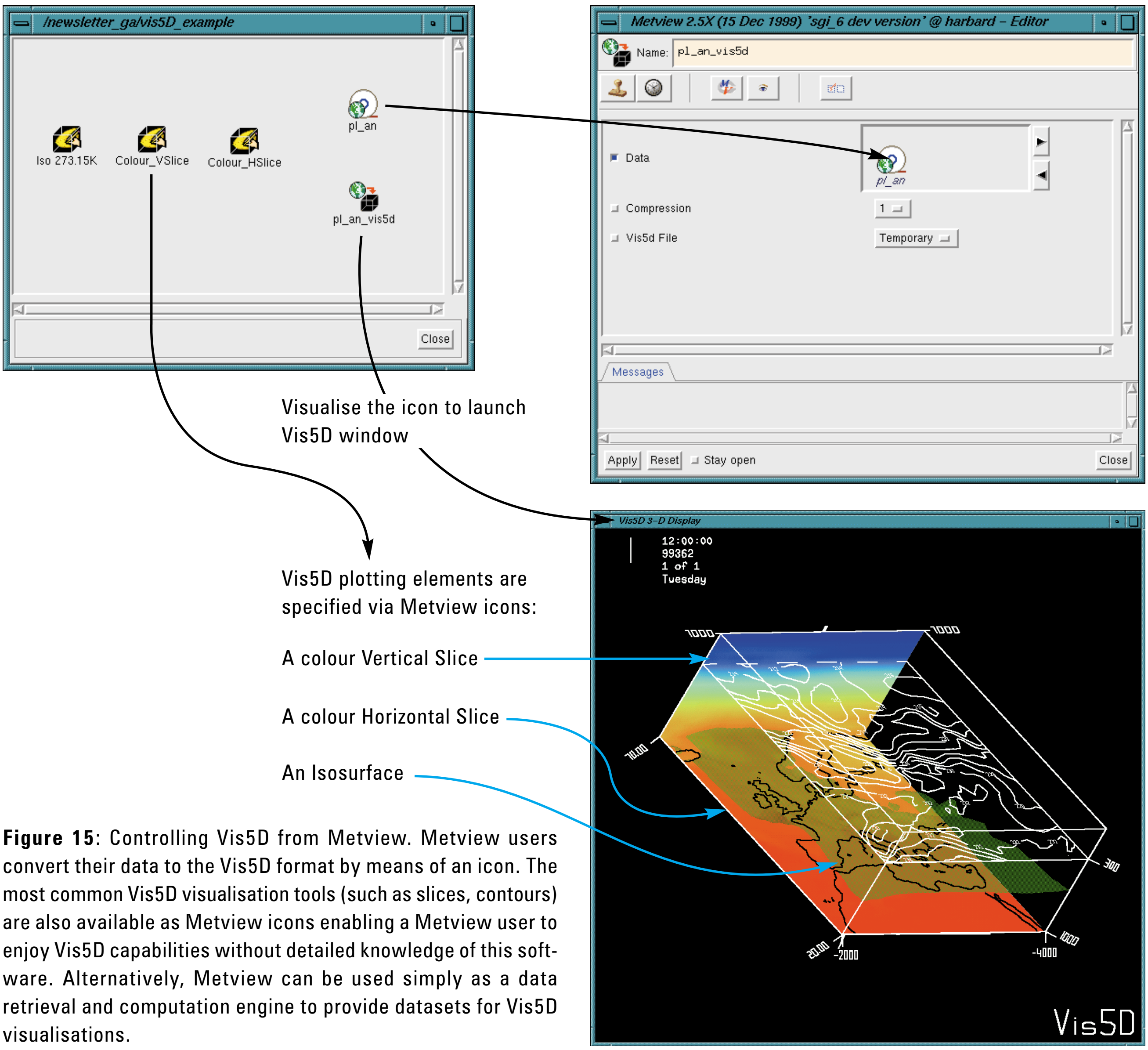

Figure 15: Controlling Vis5D from Metview. Metview users convert their data to the Vis5D format by means of an icon. The most common Vis5D visualisation tools (such as slices, contours) are also available as Metview icons enabling a Metview user to enjoy Vis5D capabilities without detailed knowledge of this software. Alternatively, Metview can be used simply as a data retrieval and computation engine to provide datasets for Vis5D visualisations.

The following data types are seamlessly handled by Metview Macro:

- strings;
- numbers;
- dates;
- lists (a vector whose elements can be of different types);
- definitions (like a list, but the elements are named);
- fieldsets (for arrays of model fields);
- satellite images;
- observations;
- geopoints (for irregularly spaced point data);
- vectors and matrices.

Variables of the types above are defined when set and there is no need for variable declarations. Users can specify operations between variables of these data types and between variables of different types (when applicable) without any effort spent in variable declarations and worrying about compatibility.

For the data types above Metview Macro provides an extensive range of built-in functions which give the user a very powerful set of processing tools; many of these functions are of a specific meteorological nature. Data entities which are cumbersome to handle in a seamless way can be easily converted (and filtered) into other more convenient types (e.g. BUFR into geopoints).

Use of functions in Metview Macro can augment significantly the power of the language. At the simplest, user-written functions can be left embedded in a program. However, Metview Macro makes it very easy for a user to build a library of functions – it is enough to save the function code files in a particular folder within the Metview system for them to be accessible as if they were system functions.

Another powerful feature of Metview Macro is the ability to use FORTRAN executables within a Macro program as if they were built-in macro functions, i.e. nothing in the macro code would identify a particular function as a FORTRAN external function. This extends immensely the scope of the macro language and enables you to make efficient use of existing resources.

A very attractive feature of Metview Macro is that it allows the user to design his/her own user interface. User interfaces are useful to provide small applications to operators who may not be conversant with the Metview Macro language. More importantly user interfaces provide generality to macro programs enabling them to accept a variety of input parameters. The basic task remains the same but it can be applied to different types of data and/or with different visual definitions.

Figure 16 shows an example of a simple user interface in action. You simply have to execute the macro program icon for the user interface to pop up. Enter the appropriate input, click the **Apply** button and the macro runs.

User interface components such as text fields, sliders, option menus, icon boxes, colour tools, etc, are defined by a set of functions known as *dialogue functions*. To implement a user interface you only need to specify the required functions and the retrieval of the input from the user interface.

Metview can run macro programs in a non-interactive or batch mode, i.e. from the command line without recourse to the user interface. To run Metview Macro in batch mode it is enough to call it from the UNIX command line, such as:

% metview -b a_macro Z 500 20000127

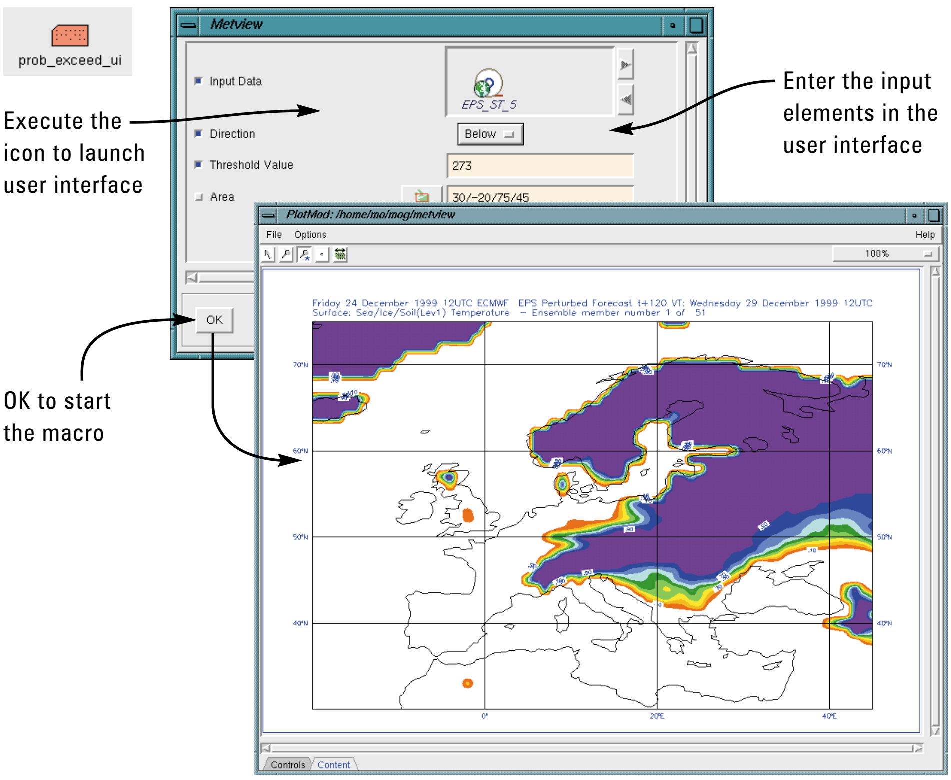

Figure 16: A macro with a simple user interface. This macro is taking an EPS set of forecast fields of surface temperature verifying at a given date, and calculating the proportion of EPS members for which temperature falls below 273 K. The macro is general in that it accepts any set of fields. EPS probabilities are the most obvious application. You may extend the macro by having it save the derived field (as well as visualising it).

Note the use of the option -b; the macro name must follow and after the macro name you specify the list of arguments the macro may need (optional).

Running macros in batch presents no special requirements, except that you are restricted to the following actions:

- saving output to a file (GRIB, BUFR, ASCII); and/or
- produce a visualisation to a file (PS or PNG) or straight to a printer.

An example of batch-mode application is in the running of operational macros (such as production of routine datasets and hard-copy plots), as the running of these jobs can be set up in shell scripts and managed with scheduling software (i.e. run during periods of minimum activity and/or network traffic).

A macro program resides in the Metview system as a **Macro** icon. As with any icon you can run it by specifying an action **(Execute Visualise Examine Drop)**. A useful feature of Metview Macro is that inside the Macro program you can retrieve its run mode, i.e. which action was specified by the user (including running in batch). This gives the user the opportunity to specify different outcomes according to the run mode the user has chosen.

A typical example is for a macro that produces a visualisation to direct the result to the screen if visualised or executed, and to a printer or PS file if run in batch mode. The user may want/need to restrict the operationality of the macro, e.g. by only allowing to run if it is dropped somewhere or by preventing it from running in batch when its only outcome is an on- screen visualisation.

Further control over a macro program run may be achieved by the user being able to read and set environment variables and to run UNIX shell scripts from within the macro program.

Rogério Bonifácio

European Union Fifth Framework Programme

Two bids to the European Commission have been successful in attracting funding for research projects co-ordinated by the ECMWF as part of the European Union's Fifth Framework Programme for Research, Technical Development and Demonstration activities in the area of Energy, Environment and Sustainable Development. The projects, ERA-40 and DEMETER, will be funded for three years, and will commence shortly.

ERA-40

The objective of the ERA-40 project is to produce, validate and undertake studies of a reanalysis covering the period since mid 1957, when major improvements were made to the atmospheric observing system in preparation for the International Geophysical Year (IGY) that took place in 1958. The project will build on the experience gained by ECMWF in carrying out ERA- 15 (an earlier reanalysis of the period 1979-1993) and on the experience of the National Centers for Environmental Prediction (NCEP) and the National Aeronautics and Space Administration (NASA) in carrying out reanalysis projects in the USA.

The reanalysis will use a three-dimensional variational (3D-Var) version of ECMWF's data assimilation system to make a new synthesis of the in-situ and remotely-sensed measurements collected during the more than 40-year period. Analyses will be produced six-hourly, with supplementary forecasts providing fields at three-hourly intervals. The basic spectral horizontal resolution will be T159, and the model grid-spacing will be close to 125 km in the horizontal, with 60 levels in the vertical located between the surface and a height of about 65 km (as is currently used for operational forecasting at ECMWF). The basic analysed variables will include not only the conventional meteorological wind, temperature and humidity fields, but also stratospheric ozone and ocean-wave and soil conditions. Additional information will be produced concerning the quality of both the observations used and the analyses generated. ECMWF's archival/retrieval system MARS will be used to store the observations, the analyses, the forecast products and the generated data. The database will be made widely available. Users without MARS access will be able to obtain a comprehensive range of data through ECMWF's Data Services, and data may also be available from some national data centres. Particular attention will be given to generating data subsets on compact media, which can then be made available to users at an affordable cost. The information produced by the project will be unique in its combination of time-range, vertical extent, variety and accuracy. An extensive documentation will also be produced to facilitate the use of the data.

The analysed products will be enhanced by short periods of higher-resolution global assimilation using 4D-Var, which will promote further exploitation of the data from major international observing experiments, such as GATE (1974), FGGE(1979), ALPEX (1982) and TOGA-COARE (1992-93). It is planned to carry out medium-range forecasts at regular intervals from the ERA-40 analyses to enable study of variations in predictability during the period.

ERA-40 will be innovative in its direct variational assimilation of raw-radiance data from satellites from 1972 onwards, and will use some new types of observation and improved specifications of sea-surface temperatures and sea-ice distributions. Analyses will be produced for overlap periods with and without each significant addition made to the observing system since the IGY. This will enable assessment and quantification of the impact of these additions and, in particular, the documentation of the benefit of the development of the satellite observing system (VTPR, TOVS, Meteosat/GOES/GMS, SSM/I, ERS and ATOVS) over the past three decades.