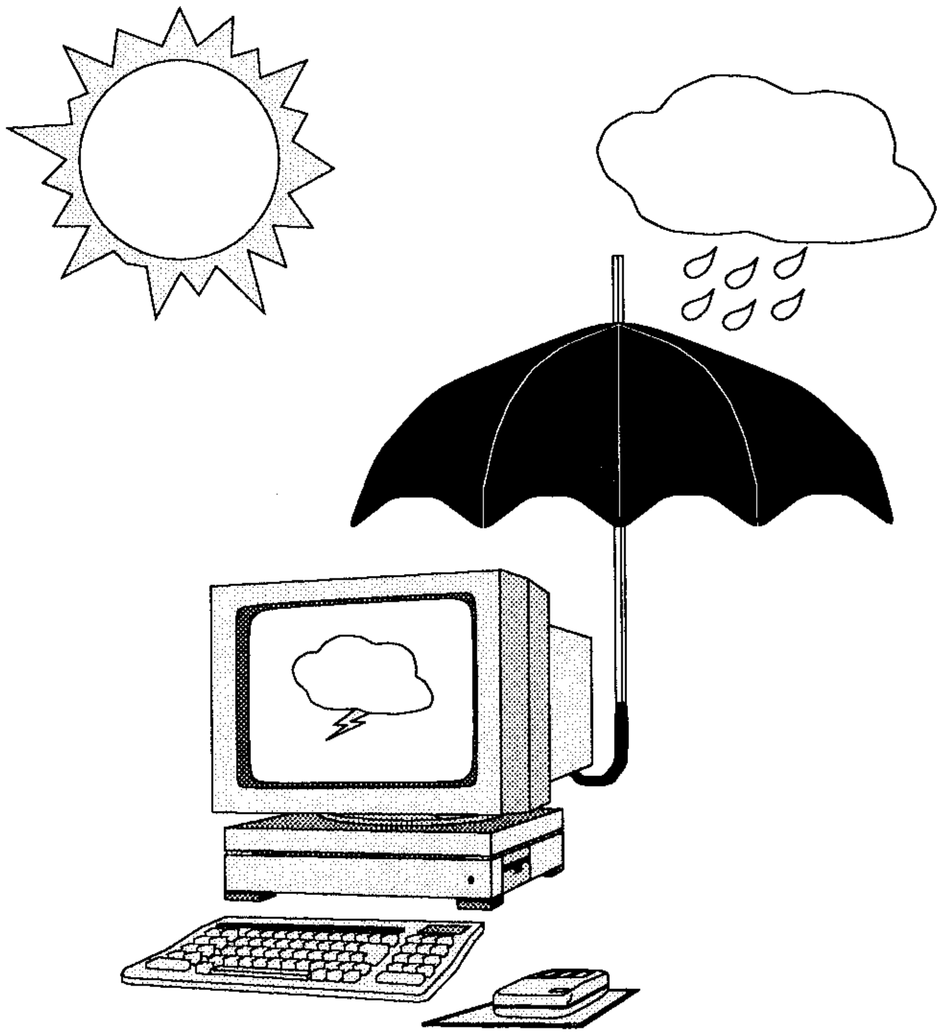


EGOWS 5

Copenhagen

20 - 23 June 1994

DMI

Contents

Foreword

Meeting programme

List of participants

Contributions:

- Written reports on Experiences with Graphical User Interfaces (GUIs) and GUI builders
- Presentations

Recommendations

Foreword

The continued interest in the European Group on Operational Meteorological Workstations is demonstrated by the fact that representatives from 18 organizations took part in this years meeting. Meteorological topics have found their way into the discussions, but technical matters still have the biggest interest. Not surprisingly, as workstation hardware is still evolving rapidly, bringing more and more computing power every year at the same nominal cost.

A look at the possible future of multiprocessing was presented on the first day of the meeting by visiting Associate Professor at Dept. of Computer Science, University of Copenhagen, Mr. Robert J. Fowler. The history and state of the art of multiprocessor processing were presented, and the results of calculations, which in many ways corresponded to NWP were put forward. Copies of the slides from the presentation were handed out at the meeting and have not been included in this report.

As many services sees a decrease in funding, but are expected to maintain or increase their number of products, automated procedures for the generation of forecasts are emerging in several places. A few experiences on such systems were presented, but no doubt there are many more to come.

In the general area, efforts are concentrating on presenting information fast and in a graphical way to give a general view from the vast amount of data available. 2D graphics can now be generated online at satisfactory speed instead of having to rely on precomputed metafiles and the like. 3D graphics is still awaiting utilization on the meteorological workstations.

I would like to take this opportunity to thank you all for your valuable contribution to the meeting. Thank you also to the staff of the Graphics Group at DMI for taking care of practical and technical matters concerning the meeting.

By kind invitation from the Austrian Meteorological Service, ZAMG, the next meeting of EGOWS is expected to take place in Vienna, June 19-22 1995.

Jacob Brock

Monday 20 June

9.30	Registration
10.00	Opening of the 5th meeting of EGOWS
10.30	'Parallel Computing Using "Commodity" Microprocessors: Possibilities, Strategies, and Limitations.' Lecture by Robert J. Fowler, visiting Associate Professor at Dept. of Computer Science, University of Copenhagen
12.15	*Lunch*
13.30	"Parallel computing" continued
14.45	*Coffee break*
15.00	Discussion on "Parallel computing"
16.00	Tour of Operations at DMI

Tuesday 21 June

Graphical user interfaces

9.00	Mr. J. Daabeck	European Centre for Medium-Range Weather Forecasts
9.20	Ms. M. F. Voidrot	MeteoFrance *France*
9.50	Mr. M. Pogoda	Deutscher Wetterdienst *Germany*
10.05	Mr. P. Trevelyan	Meteorological Office *United Kingdom*

WMO Distributed Databases project

10.30	Mr. D. McGuirk	World Meteorological Organization
	Coffee break	

Meteorological applications

11.00	Mr. O. Spaniel	Slovensky Hydrometeorologicky Ústav *Slovakia*
11.30	Mr. M. Göstl	Zentralanstalt für Meteorologie und Geodynamik, *Austria*

12^{00}	Mr. Å. Horváth	A Magyar Köztársaság Meteorológiai Szolgálata, *Hungary*
12^{30}	*Lunch*	
13^{30}	Mr. T. Madsen	Danmarks Meteorologiske Institut *Denmark*

General developments

14^{00}	Mr. C. Lemcke	Koninklijk Nederlands Meteorologisch Institut, *Netherlands*
14^{30}	Mr. A. Kaiser	Schweizerische Meteorologische Anstalt *Switzerland*
15^{00}	*Coffee break*	

Demonstrations

15^{15} Demos: Amt für Wehrgeophysik on SGI
MeteoFrance on SUN
Deutscher Wetterdienst on SGI
Slovak Hydrometeorological Institute on HP Apollo
Danish Meteorological Institute on SUN

Walk through Copenhagen

Wednesday 22 June

General developments (continued)

9^{10}	Mr. J. Hamilton	An tSeirbhís Mheitéareolaíochta *Ireland*
9^{40}	Mr. L. Gerard	Institut Royal Météorologique de Belgique, *Belgium*
10^{10}	Ms. E. Icart	MeteoFrance *France*
10^{45}	*Coffee break*	
		France
11^{00}	Mr. M. Pogoda	Deutscher Wetterdienst *Germany*

11:15	Mr. W. Lipa	Zentralanstalt für Meteorologie und Geodynamik, *Austria*
11:45	Mr. Å. Horváth	A Magyar Köztársaság Meteorológiai Szolgálata, *Hungary*
12:00	Mr. P. Dahlén	Swedish Meteorological and Hydrological Institute, *Sweden*
12:30	*Lunch*	
13:45	Mr. P. Trevelyan	Meteorological Office *United Kingdom*
14:00	Mr. K. Christensen	Danmarks Meteorologiske Institut *Denmark*
14:30	Mr. J. Daabeck	European Centre for Medium-Range Weather Forecasts
15:00	*Coffee break*	

Demonstrations

15:15 Demos:
Amt für Wehrgeophysik on SGI
MeteoFrance on SUN
Deutscher Wetterdienst on SGI
Slovak Hydrometeorological Institute on HP Apollo
Danish Meteorological Institute on SUN

Social evening

Thursday 23 June

9:10 Discussion on

- recommendations
- experiences with GUIs
- WMO DDB

10:30 *Coffee break*

Next meeting

12:00 Closing

Participants at EGOWS 5

Mr. Peter Trevelyan UK Met Office Central Forecasting Division London Road Bracknell UK - Berkshire RG12 2SZ England Phone: +44 344 854882	**Mr. Jens Daabeck** ECMWF Operations Department Shinfield Park Reading UK - Berkshire RG2 9AX England Phone: +44 734 499375 Fax: +44 734 869450 Jens.daabeck@ecmwf.co.uk
Ms. Angela Smith UK Met Office Central Forecasting Division London Road Bracknell UK - Berkshire RG12 2SZ England Phone: +44 344 856408 Fax: +44 344 854412	**Mr. José J. S. Marques** Instituto de Meteorologia Rua C Aeroporta de Lisboa P - Lisboa Portugal Phone: +351 3511 8483961 Fax: +351 3511 802370
Mr. Luc Gerard Institut Royal Meteorologique Dept. Applied Meteorology Avenue Circulaire 3 B - 1180 Brussels Belgium Phone: +32 2 373 05 64 Fax: +32 2 375 50 62 Luc@magics.oma.be	**Mr. C. Lemcke** KNMI PO Box 201 NL - 3730 AE De Bilt Netherlands Phone: +31 30 206356 Fax: +31 30 210407 lemcke@knmi.nl
Mr. David McGuirk WMO 41 Avenue Guiseppe Motta Case Postale no. 2300 CH - 1211 Genève 2 Switzerland Phone: +41 22 730 8241 Fax: +41 22 733 0242 D.McGuirk@omnet.com	**Mr. Friedhelm David** Amt für Wehrgeophysik Mont Royal D - 56841 Traben-Trarbach Germany Phone: +49 6541 18 282 Fax: +49 6541 18 296
Mr. James Hamilton Meteorological Service Glasnevin Hill EIR - Dublin 9 Ireland Phone: +353 1 8375780 Fax: +353 1 8375557 duj@ecmwf.co.uk	**Mr. Reinhard Trapp** Amt für Wehrgeophysik Mont Royal D - 56841 Traben-Trarbach Germany Phone: +49 6541 18 444 Fax: +49 6541 18 296

Ms. Marie F. Voidrot
MeteoFrance
SCEM/TTI/DEV
42. Av. G. Coriolis
F - 31057 Toulouse Cedex
France
Phone: +33 61 07 81 27
Fax: +33 61 07 81 09
marief@meteo.fr

Mr. Ales Poredos
Hydrometeorological Institute of Slovenia
Vojkova 1B
61000 Ljubljana
Slovenia
Phone: +386 61 313 583
Fax: +386 61 320 466
poredos%hmljk@uni-lj.SI

Ms. Elisabeth Icart
MeteoFrance
SCEM/TTI/DEV
42. Av. G. Coriolis
F - 31057 Toulouse Cedex
France
Elisabethicart@meteo.fr

Mr. Wolfgang Lipa
ZAMG, Vienna
Hohe Warte 38
A- 1190 Vienna
Austria
Phone: +43 1 36 44 53
Fax: +43 1 36 51 233
Lipa@edvz.zamg.ac.at

Mr. Peter Dahlén
SMHI
Folkborgsvägen 1
S - 601 76 Norrköping
Sweden
Phone: +46 11 158000
Fax: +46 11 170207
pdahlen@smhi.se

Mr. Manfred Göstl
ZAMG, Vienna
Hohe Warte 38
A- 1190 Vienna
Austria
Phone: +43 1 36 44 53
Fax: +43 1 36 51 233
goestl@edvz.zamg.ac.at

Mr. Michael Pogoda
Deutscher Wetterdienst
Wetteramt Potsdam
Michendorfer Chaussee 23
D - 14473 Potsdam
Germany
Phone: +49 331 316 527
Fax: +49 331 316 292
pogoda@w3-map.wa-potsdam.dwd.d400.de

Mr. Åkos Horváth
Hungarian Meteorological Service
Kitaibel Pál u. 1.
H - 1024 Budapest
Hungary
Phone: +36 1 212 2699
Fax: +36 1 212 5153
H10257Hor@ella.hu

Mr. Oldrich Spaniel
Slovak Hydrometeorological Institute
Jeséniova 17
833 15 Bratislava
Slovak Republic
Phone: +42 7 374 052
Fax: +42 7 374 374
ol@shmi.sk

Mr. Karel Pesata
Czech Hydrometeorological Institute
Na Sabatce 17
143 06 Praha - Komorany
Czech Republic
Phone: +42 2 4016503
Fax: +42 2 4010800
pesata@CHMI.CZ

Mr. André Kaiser Swiss Meteorological Institute Krähbühlstrasse 58 Postfach CH - 8044 Zürich Switzerland Phone: +41 1 256 91 11 Fax: +41 1 256 92 78 `aka@sma.ch`	**Mr. Ole Hansen** DMI Lyngbyvej 100 DK - 2100 København Ø Denmark Phone: +45 39 15 72 66 Fax: +45 39 29 12 12 `oh@dmi.min.dk`
Mr. Thomas Madsen DMI Lyngbyvej 100 DK - 2100 København Ø Denmark	**Mr. Torben Strunge Pedersen** DMI Lyngbyvej 100 DK - 2100 København Ø Denmark Phone: +45 39 15 72 62 Fax: +45 39 29 12 12 `tsp@dmi.min.dk`
Mr. Knud Erik Christensen DMI Lyngbyvej 100 DK - 2100 København Ø Denmark Phone: +45 39 15 72 65 Fax: +45 39 29 12 12 `kec@dmi.min.dk`	**Mr. Jacob Brock** DMI Lyngbyvej 100 DK - 2100 København Ø Denmark Phone: +45 39 15 72 64 Fax: +45 39 29 12 12 `jbb@dmi.min.dk`
Mr. Jan Borris DMI Lyngbyvej 100 DK - 2100 København Ø Denmark Phone: +45 39 15 72 48 Fax: +45 39 29 12 12 `jbo@dmi.min.dk`	

Contributions

Written reports on

Experiences with Graphical User Interfaces (GUIs) and GUI builders

prepared for

EGOWS 5

The Met. Office

United Kingdom

Experiences with GUIs in the Horace Project

1) At the start of the Horace Project in 1991, it was decided that we would use Graphical User Interfaces (GUIs) based on the Motif Toolkit. At that stage it seemed to us that Motif was becoming the standard for Unix platforms. It was also apparent that Motif has some similarities to Microsoft Windows which our users are more likely to be familiar with. It was also considered possible that we may buy in some commercial software packages to use in Horace and these would most probably use Motif as well.

2) It was soon obvious that creating Motif GUIs from scratch would be a difficult task, so an evaluation of a number of GUI builders was undertaken. These included X Designer from IST, Teleuse from Telesoft, Interface Architect from Hewlett Packard and Alex from Alex Technologies.
Alex was ruled out because of the cost of run time licences. Teleuse was considered to be expensive and rather overpowering in its complexity. X Designer and Interface Architect were very similar in price, but at the time of the evaluation X Designer was clearly a much easier and more intuitive tool to use. This opinion was confirmed during discussions with ECMWF who were already using X Designer and who appeared to be very pleased with it. It was decided that we would purchase two seats of X Designer.

3) Our experience of using X Designer has been very positive. Of all the tools purchased for the Horace Project, X Designer has by far been the most successful. There have been some minor problems with the earlier releases but most of these have been fixed in version 3.20
The ease and rapidity of developing prototype GUIs that we can demonstrate to the users has been very useful. The ability to drag and drop and cut and paste widgets or assemblies of widgets and immediately see what the dialogs look like, saves an enormous amount of time. Of course once the layout of the GUI has been finalised the hard work of coding the callbacks to execute the appropriate actions must take place. In order to make this somewhat easier, we have developed a number of useful library routines that can be used in our GUIs.

4) In the early days of the Horace Project when there was little experience of creating GUIs a certain amount of experimentation took place. This resulted in a lack of standardisation between different GUIs. It was also recognised that we had not made enough use of things such as pulldown menus, accelerators and mnemonics. To remedy these problems we have now developed a set of standards that all new GUIs must adhere to and we are also upgrading our existing GUIs to meet those standards.
One policy that we have adopted is not to hard code colours into our GUIs but to have them use the colours set by Hewlett Packard's Visual User Environment (VUE). In the next major release of Horace our Main Menu will be replaced by a customised VUE Front Panel. This will give the users access to Style Manager which will allow them to adjust things such as colour to their own preferences which is of particular interest to people who are colour blind.

R.P.Townsend
25/05/94

GUI Standards Document

Accelerator Keys

1)The use of accelerators is optional. In general they should be employed for frequently used actions where they will facilitate speedier performance.

2)The following letters are reserved for specific actions:

<Ctrl> D	Display
<Ctrl> O	Open
<Ctrl> P	Print
<Ctrl> S	Save
<Ctrl> X	Exit

3)Using X-DESIGNER to set accelerators and mnemonics:

Accelerators and mnemonics can be set by using the *Keyboard* resources for the widget as follows:

e.g. for a Display button

RESOURCE	ENTRY
Accelerator	Ctrl<Key>D
Accelerator Text	<Ctrl> D
Mnemonic	D

Arrow Buttons

1) When creating text widgets that can be incremented or decremented the up and down arrows must be placed one above the other on the right hand side of the text widget. Set the border and highlight core resources to 0.

2) Scrolled arrows must have auto scrolling where ever possible.

Buttons

1) Softcopy and hardcopy options must be separate pushbuttons.

2) Standard pixmaps for softcopy and hardcopy must be used.

3) Hardcopy devices will be selected using an option menu widget.

4) The preferred ordering of pushbuttons is softcopy, hardcopy, exit.

Close Handler

1) All gui's must use XhCloseHandler to register a callback to respond to a close command from the window manager. The callback must execute appropriate actions such as killing child processes and updating files.

Colours

1) Colours must not be hardwired.

2) All toggle buttons must have a call to XhSetToggleSelectColor in their pre-manage code preludes. The toggle button's variable name is passed as the only argument to the call. Using XhSetToggleSelectColor ensures the select colour changes dynamically if the user adjusts the background colour using VueStyle.

3) See the Pixmaps section for the list of colours to be used for pixmaps.

Environment Variables

1) Do not create unnecessary environment variables.

2) For printing use either of the following

    TEXT_PRINTERS=ljet1:pjet1text
    GRAPHICS_PRINTERS=djetd1:pjet1:A0

Error Logging

1) To aid debugging all GUI's must write clear messages to the error log.

2) Severe errors must be reserved for notifying supervisors of problems that need their attention.

3) When an application dies, a pop up must be displayed telling the user this. On confirmation of the dialog the GUI must also die.

Layout

1) The layout of the GUI controls must be as ergonomic as possible, to minimise mouse movements.

2) When creating a GUI place the options on the left of the GUI and the output area on the right.

3) Where possible the selection menu and the output menu should be combined into a single dialog.

4) The menus should be as large as possible, but must not obscure the main menu.

5) On large dialogs group widgets into logical areas and surround them with a frame.

Menu Bars

1) All top level windows must have a Menu Bar.

2) Do not use pixmaps on the menubar.

3) If the only menu item on a menubar is HELP and the item only has one topic, then you can dispense with the menubar and have a help pushbutton on the dialog.

4) The standard menus for the menu bar should be: File, Edit, View, Options and Help. File and Help are mandatory but Edit, View and Options can be used only in appropriate circumstances.

5) An application specific menu (e.g. "Insert") may be added to the menu bar in circumstances where the contents of the new menu do not readily fit into a standard menu. In this case the menu should be placed after the standard menus but before the Help menu.

6) The standard mnemonics for the menu bar are File, Edit, View, Options and Help.

7) The **FILE** menu

The items in the File menu should be grouped in the following way and those groupings should also be in the following order :

- actions about the application e.g. New, Open, Save.
- actions transferring changes made to a storage medium e.g. Copy, Rename, Delete.
- actions concerned with transmission e.g. Display, Print.
- application specific actions as required.
- action to exit e.g. Exit.

Exit is the only mandatory item .

The groupings, where used, should be made distinct from each other by means of Separator widgets.

The number of items in the File menu should not be excessive.

8) The **EDIT** menu

The Edit Menu should contain those actions that modify the contents of the data that the application is currently dealing with e.g. Copy, Cut, Paste etc.

9) The **VIEW** menu

The View menu should contain those actions that allow users to alter the way data is displayed by the GUI e.g. All, Partial, By Date etc.

10) The **OPTIONS** menu

The Options menu should contain those items that allow users to customize different aspects of the GUI e.g. Preferences, Colours etc.

Mnemonics

1) The use of mnemonics is optional EXCEPT on the menu bar itself where they are mandatory. In general they should be employed for frequently used actions where they will facilitate speedier performance.

2) To set mnemonics in X Designer, see the section on Accelerators.

Pixmaps

1) The following is a list of the only colours that should be used when creating pixmaps. This list contains 17 colours, 5 of which are grey. If you want to add any more colours to this list please consult the rest of the team first.

AntiqueWhite
Black
Blue
DarkOrange
DarkSeaGreen (a light green)
DeepSkyBlue2 (a light blue)
Firebrick (a dark red)
ForestGreen (a dark green)
Gray40 (the darkest grey)
Gray50
Gray60
Gray70
Gray80 (the lightest grey)
Purple
Red
White
Yellow

2) Pixmaps must be generated into a separate header file to the rest of your GUI code.

3) All top level windows must have a two colour pixmap for iconising.

4) The top level window must contain icon help, for all pixmaps used.

5) When pixmaps have been agreed and created the pixmap is attached to your GUI. So that everyone else can use your pixmap, you create a file, containing your pixmap, with the extension .xpm and put it in the directory "/usr/horace/../source/gui/x_designer/xpm".

6) Great care must be taken when choosing an icon to represent an action.

Popups

1) Popup dialogs must be used
- to warn the user, when saving, if a file will be overwritten.
- to warn the user, when quitting, that data will not be saved
- to inform the user that the application has died .

2) When a confirmation dialog is popped up, a user must not be allowed to continue with the main option menu until the confirmation dialog has been acknowledged. This can be done with XDesigner by setting the BulletinBoard resource dialog style to "primary application modal".

Source Code

1) The source code for each gui will be kept in a seperate directory under `../source/gui/motif/`.

Status Bars

1) All top level windows must have a status bar.

2) The status bar must be enclosed in a frame and placed at the bottom of the window.

3) Complex secondary windows must also have a status bar.

4) There must be enough messages displayed to the user, to keep the user aware of the current state of the GUI and application, making use of the hour glass cursor where appropriate.

5) GUI must display the results of an action e.g. request to print failed.

Testing

1) All GUI's must be thoroughly tested with their application. All possible selection options should be tested and the output should be compared with that produced by other systems, such as ODS and TSO where available.

Xh Library

1) The latest source modules for libXh are now in /usr/horace/latest/source/gui/motif/libXh

the latest makefile is in /usr/horace/latest/source/make
and the include file is in /usr/horace/latest/source/include.

2) When a new function has been developed for libXh it must be reviewed by the GUI Team Leader before being moved to /usr/horace/latest.

3) All GUI team members must regularly test their software with the version of libXh on /usr/horace/latest. If you have any problems report them to the developer of the function concerned or the GUI Team Leader.

4) Always remember to update the file templates when making changes to modules. If you are changing an old module that used the original template then it would be a good idea to convert it to the latest type.

5) Please ensure that you use the correct style for Pre and Post conditions. See XhDisplayHelp.c for an example of how they should be done.

6) Always make sure that all the functions in libXh, as well as functions that use libXh have a #include <Xh.h> statement. This will mean that function prototypes are checked.

7) All functions in libXh will have an Island Write document in `/usr/horace/docs/gui` describing their purpose, usage etc. The documents will be based on the template `/usr/horace/docs/gui/XhTemplate.doc`. The filename of document will consist of the name of the function with a .doc extension.

8) Ensure that you do not duplicate an existing function in the Xh library.

9) When creating your own subroutines or callbacks try to make them as general as possible so that they can become library routines.

Danmarks Meteorologiske Institut

Denmark

Considerations on choosing an X Development Platform at DMI

X does not define any standards or guidelines for "Look & Feel", ie. the graphical structure of the user interface and the appearance of the graphical primitives (button, menues etc.)

Two software vendors, SunSoft and OSF (Open Software Foundation), have defined standards for Look & Feel (OpenLook and Motif), and libraries to support application development have been implemented.

Until recently (about a year ago) the Weather Service Division at DMI have been using OpenLook, but have chosen to use Motif in the future. The main reason for this decision is to ease the task of porting our applications to platforms other than Sun. Shortly after this decision was made, Sun announced that they had entered the COSE (Common Open System Environment) group, which means that SunSoft in the future will use the COSE desktop, which uses Motif Look & Feel, as the default desktop with OpenWindows. So at the moment it looks as if OpenLook has lost the battle.

For developing XView based applications, we have used the GUI builder DevGuide (Developers Guide) from Sun, and we have been quite satisfied using that. After changing to Motif, we developed a few applications "from scratch", and found that a lot of time could be saved using a GUI-builder, so we spent some time getting an overview of various GUI-builders before deciding. In the remaining part of this report, we describe what we learned through this process.

Comments on various GUI-builders

ViewCenter (Centerline Software):

Some GUI-builders (like ViewCenter) generate applications which support both the OpenLook and Motif standards for Look & Feel. The "style" is selected at run-time (start the program with option -motif or -openlook). This seems to be a very clever idea, however there are a number of technical reasons to be cautious using this kind of tool.

To be able to implement this functionality, the vendor has invented an intermediate level, in which all graphical primitives (widgets) can be presented in conformance with both supported standards for Look & Feel. This impose some limitations on the available widgetset (greatest common denominator), and, in our opinion, the resulting applications do not completely conform to the selected Look & Feel. Furthermore programmers have to learn a new API (Application Programming Interface) and we are dependent on the future availability of the product.

Open Interface (Neuron Data):

An example of a "Platform Independent" tool. The term "Platform Independent" means that the code generated can be compiled to run on non-X platforms (ie. MacIntosh or MS Windows). Such products have obvious advantages, but most of the considerations mentioned above, applies here as well. Furthermore this product is very expensive, and requires "run-time" licences for the produced binaries.

Teleuse (TeleSoft):

Teleuse implements a UIMS (User Interface Management System). The code generated by a UIMS is structured in such a way that the definition of the static user interface (what is shown on the screen) is completely separated from the semantic part (ie. what happens when you interact with the interface). The theoretical advantage of this separation is that the graphical part of the application, in principle, can be replaced - either by another graphical interface, or by a simple tty-driven interface. However this is, in our case, not very interesting, furthermore is this product quite expensive.

Builder Xcessory (ICS):

Generates pure C-code. It is prepared for integration with the debugger "CodeCenter" from CenterLine (C-interpreter, incremental linking, etc). Together these products offers an integrated tool with which you can generate code, test and debug.

We spend about a week evaluating Builder Xcessory/CodeCenter, and found two rather serious bugs in the product, which led us to conclude that the product was immature at the time.

XDesigner (IST):

XDesigner is quite similar to Builder Xcessory. It generates pure C-code (or C++) and it integrates with CodeCenter. XDesigner was recommended to us for by people who had been using it for rather large projects

UIM/X (Visual Edge):

As TeleUse is, a UIMS, however UIM/X generates pure C/Motif code, ie. does not introduce any intermediate levels. UIM/X has an integrated C-interpreter so that both interface code and callback functions can be tested and debugged from within UIM/X.

We did not ourselves test UIM/X, but according to people who have been using UIM/X for large projects. UIM/X is well functioning when developing rather small applications. However it gets slow and unflexible when working with large applications - the "turnaround time" gets unacceptable, probably due to a non-incremental interpreter.

Tcl/Tk (John Ousterhout):

Tcl/Tk is actually not a GUI-builder, but a language (Tcl) and a windowing toolkit (Tk). Tcl itself is an interpreted language, and Tk is toolkit implementing a number of "widgets". We have implented a few, small, applications using Tcl/Tk, and have found it very usefull.

Tcl/Tk belongs in the "public domain", and we have used it on both SunOS4.x and Solaris.

Tcl/Tk is not suited for handling images.

Conclusions:

One of our primary demands to a GUI-builder is that it must generate pure C/Motif code since this enables us to continue development of the code even when/if the vendor for some reason disappears from the market. This leaves us with Builder Xcessory, UIM/X and XDesigner. Based on the facts/experiences described above, we have chosen XDesigner.

So far only a few Motif applications have been developed using XDesigner and Motif from SunSoft (SDK), so our current experience is limited. However it seems to be a robust tool, and we have had no problems with the generated code.

Jan Borris
DMI Weather Service Division
Graphics Group

Swedish Meteorological and Hydrological Institute

Sweden

Experience from GUIs and GUI builders at SMHI

The purpose of this document is to describe SMHI's experience from Graphical User Interfaces and development tools, GUI builders, for these GUIs. This document will only discuss standard GUIs.

Platforms and GUIs

The table below shows the GUIs that have been used or are now used at SMHI.

Platform	OS	GUI	Comment
Macintosh, Power Mac	Mac OS	Mac GUI	For media production and distribution.
PC	DOS/MS-Windows	MS-Windows 3.x	Satellite, radar, model-output presentation and forecast-production. Customer systems
VAXstation	VMS	DEC Windows	System management and development. Replaced by X-Windows/Motif 1992
VAXstation	VMS	Motif	System management and development.
DECstation	Ultrix	Motif 1.1	System management and development, research, operational.
DEC Alpha AXP	OSF/1 2.0	Motif 2.0	GIS, sys. man/dev, research, new satellite system, IceMap
DEC Alpha AXP	openVMS 1.5 AXP	Motif 1.2	System management
Silicon Graphics Indy	IRIX 5.2	Indigo Magic Motif 1.2	Hydrodynamic model calculations (Phoenics) and visualization (Explorer). Visualization of observations.

In the future SMHI will also have a variety of platforms and GUIs. The GUIs are quite similar today, and are likely to get even more similar in future versions. Due to that, the important thing is not to work with one specific GUI, but to be able to carry out the meteorologists' task at one site with one computer.

The following chapters describe our experience from the GUIs in the development of meteorological, hydrological or oceanographical applications. Each person's opinion about different GUIs seems to depend on which they are using, or they first were introduced to. Due to that the following will focus on advantages or disadvantages with different GUI builders, rather than different GUIs.

Mac OS

Applications

The operational MAC applications are based on commercial standard applications with different adaptations. Most of the adaptations have been done by external computer consultants.

Adobe Illustrator and Aldus Freehand are used for production of newspaper material. A product generator (VAX/VMS based) generates products using product templates. Then the forecaster uses Illustrator/Freehand manually to add information that the product generator can't insert.

For the TV production we are working manually, using tools such as PhotoShop and MacroMind Director. To decrease the amount of manual work, parts of the production have been automated using QuickKeys and similar tools.

GUI Tools

No GUI builder has been used.

DOS/MS-Windows

Applications

MS-Windows applications have been developed by SMHI for internal use as well as for external customers. At SMHI we are using Windows applications to produce forecasts (text, limited graphical functionality), and to visualize radar and satellite images and model output (post processed products).
Our customers use our Windows systems to receive and display different types of products from SMHI, for example observations, textual forecasts, satellite, radar, thunderstorm observations and forecasts. Usually they receive the products using an ordinary modem. Some of the applications have facilities for running local models (for example hydrological or environmental), using input received from SMHI, or entered manually.

GUI Tools

Microsoft Visual Basic (VB)

Most of the Windows applications are developed with this tool. With VB it is very simple to draw your windows, add your objects (menus, textfields, pushbuttons, pictureboxes, etc.) and write the code that is associated with different events. The code is not written as callbacks as in many other GUI builders, but directly in a structured, "semi object oriented" version of BASIC.

Advantages
- High productivity
- A lot of 3rd party objects (widgets)
- Simple to learn
- Extendible with subroutines or own objects
- The perfect tool for prototyping

Drawbacks
- Bad performance when writing complicated code in VB
- All Windows events are not available as Visual Basic events.

Microsoft Visual C++ (VC)

VC has only been used for a few months, and no operational application has been created. With the tool you draw your interface quite similar to VB. The code is written as callback routines from the different objects events. C or C++ is available.

Advantages
- A lot of functionality available in Microsoft Foundation Classes
- Higher performance than VB
- A lot of 3rd party classes
- Built in "Application Wizards" guide you when you build applications.

Drawbacks
- Difficult to learn
- Lower productivity

Gupta SQLWindows

The main difference between SQLWindows and VB is the language that is called SAL - SQLWindows Application Language. SAL is more object oriented then VB, since it has classes and inheritance. SQLWindows is powerful when you build client/server database applications (SQL). The tool has only been used at SMHI since february 1994. Gupta has announced that they will release a Macintosh version and later on a Motif version. Extension to the language can be written in C, C++, Assembler or FORTRAN.

ULTRIX using X-Windows/Motif

Applications

The user interface is well designed and self explained. We have not come across any special shortcomings. An advantage with X-Windows/Motif is the ability to open and use windows across platforms.

We have developed some systems that use Motif, like

Metgraf - for plotting meteorological fields, cross sections and observations

Monitor - for monitoring the HIRLAM model (not finished)

Trajectory - for defining input date for trajectory calculations and presenting the result on a map

Mesoscale analysis - a development tool for displaying data from radar, satellite, synoptical observations, and analysis. The system is used to identify possible problems in the analysis. It can do loops, overlay observations, images and analyses and start remote analyses.

GUI-Tools

DEC VUIT

VUIT is the same kind of tool as X-designer, UIM/X and TeleUse, but with less functionality. Due to that fact, DEC has decided not to develop the tool anymore.
Without knowledge of Motif, the widget tree and the relations between widgets, building a user interface using VUIT is difficult. On the other hand, if you have that knowledge, VUIT is quite handy to use and you can quickly design a user interface.
SMHI will replace it with a better tool.

Ingres 4GL

Ingres 4GL works very much like VB. You draw your layout of the window, and write code in the tools own language. The tool is of course specialised in writing database applications for Ingres. The main advantage is that the applications are portable (MS Windows, X-Windows). Since Ingres is not our main RDBMS, it has only been used for one application (hydrology).

OSF/1 using X-Windows/Motif

Applications

Our new Satellite system, which will be operational in September 1994, runs on the DEC Alpha platform with OSF/1 Motif. The system is implemented by Innovativ Vision. They are using X-designer as their GUI builder.

IceMap - implemented by VTT in Finland - is an application for computer aided design of ice charts. Satellite images can be displayed as background information when drawing the ice chart. The drawing tool is object oriented and handles normal drawing symbols (lines, ovals, text) and special ice-map-symbols. In addition to the ice chart for the customers, the application will also produce input to HIRLAM and to the ice drift model (late 1994). The GUI is designed using UIM/X.

AU-system implements the supervision tool for the NORDRAD system that will be delivered in September 1994. The prototype was implemented with Visual Basic (MS-Windows) and the operational version will be implemented using X-designer.

OSF/1, Motif is very likely to be the future Workstation platform at SMHI.

GUI Tools

This far no applications have been developed for OSF/1, Motif at SMHI. SMHI will during autumn 1994 select X-designer, UIM/X or TeleUse as our GUI builder for X applications.

Presentations at the meeting

ECMWF

Metview/ws 1.1

An application for data processing and visualisation

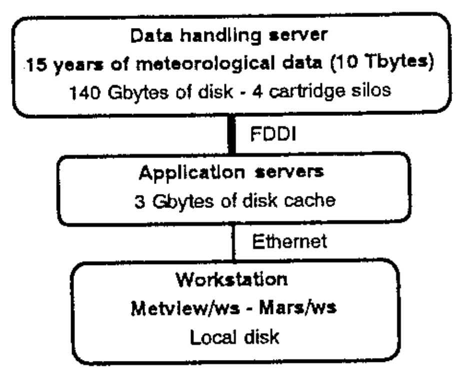
Overview

- Introduction
- Metview/ws concept
- Features
- System architecture
- Plans
- Summary

Metview/ws

- Data access, data manipulation and visualisation of fields, images and observations
- Application modules
- Unix workstations
- Motif based graphical user interface
- Distributed processing
- Full functionality performed by macros
- Batch support

ECMWF data archive Mars



User requirements

- Complete control over data display
- Interactive user interface
- Mathematics on fields
- Full functionality performed by macros
- Batch support
- Specific products
- Keep global definitions
- Help facility

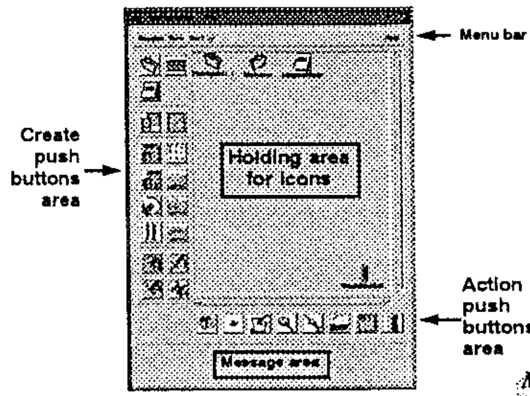
System requirements

- Unix workstations network
- Motif based graphical user interface
- Distributed processing
- Portability
- Data access

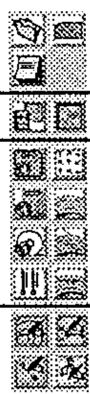
The Metview/ws concept

- User creates icons
- Icons hold definitions
  - Mars retrieval
  - Magics specification
  - Plot windows layout
  - Other
- User performs actions on icons
  - Visualise
  - Edit
  - Other

Metview/ws main window

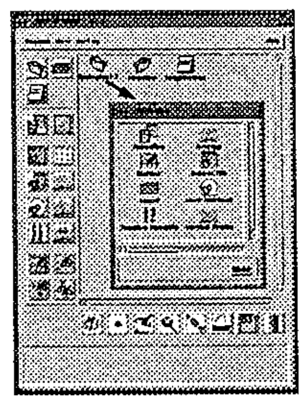


Create push buttons area

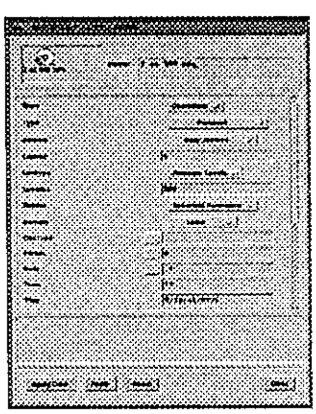


Folder	Macro
Notes	
Animation window	Plot window
Data file	Coloured wind
Compute formula	Cross section
Mars retrieval	Profile
Relative humidity	Average
Contour	Obs plot
Wind plot	Coastlines

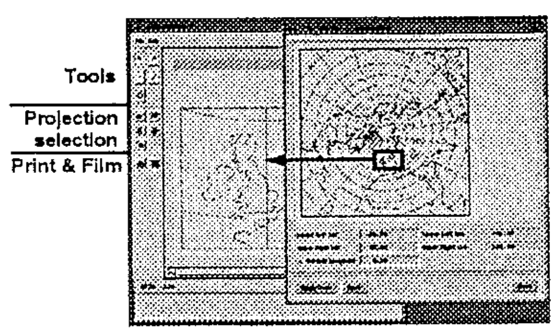
Folder



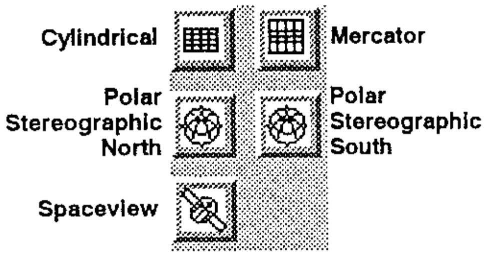
Editor window for Mars retrieval



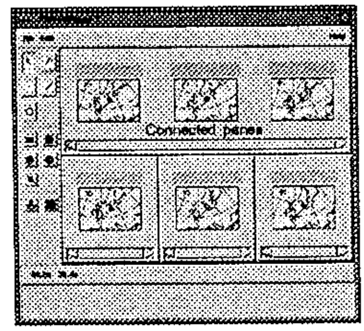
Plot window and map editor



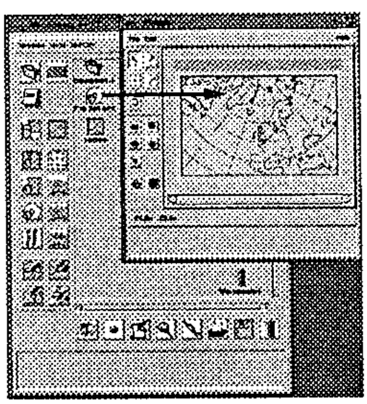
Plot window projections



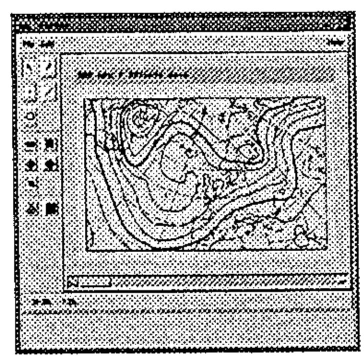
Plot window with 2 x 3 panes



Drag and drop



Plot window with one pane



Holding area

Icons can be operated upon in the following ways:

- Action push buttons

  Execute Edit Link Save

  Visualise Examine Print Delete

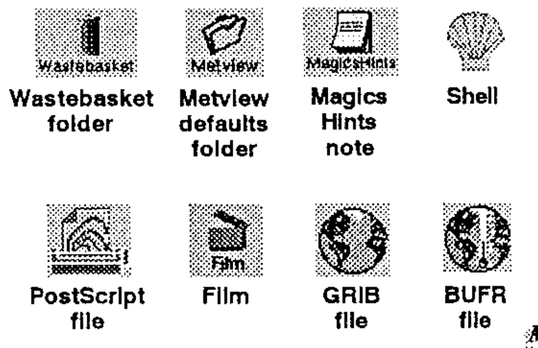
- By pop-up menus
- Drag and drop
- Double click

Holding area

The colour of the name of an icon shows its state:

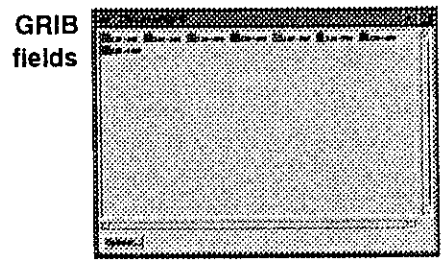
- Black - Default
- Yellow - Busy
- Green - Ready
- Red - Error

Additional icons in the holding area

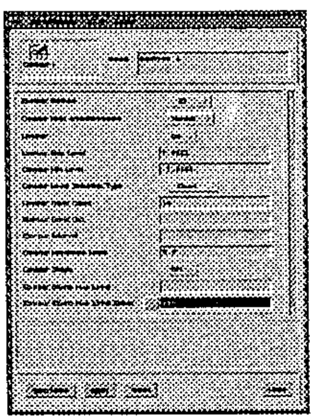


Examine action

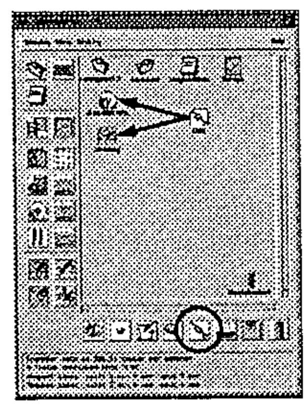
GRIB file



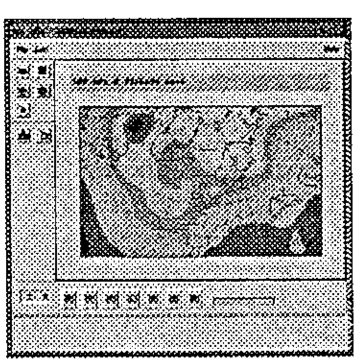
Editor window for Magics parameters



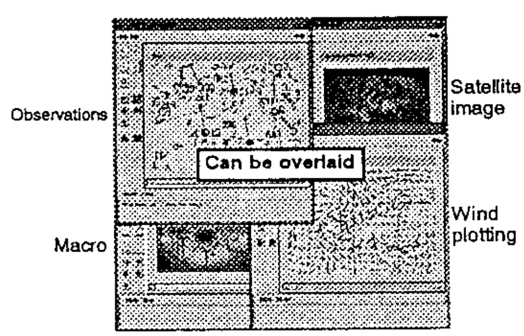
Link



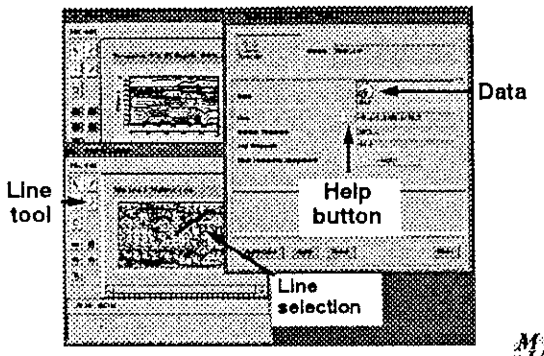
Animation window



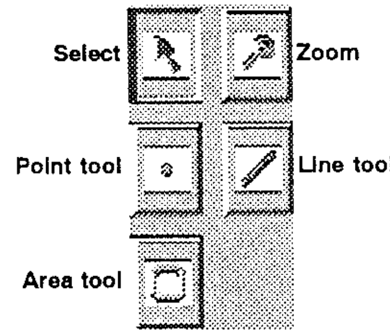
Visualisation examples



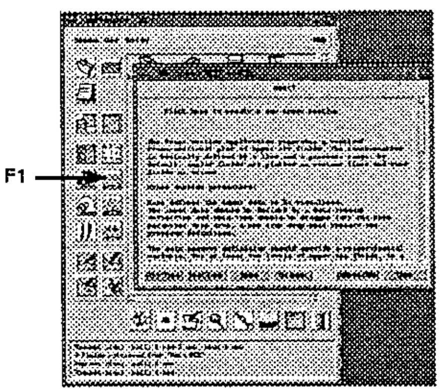
Cross section



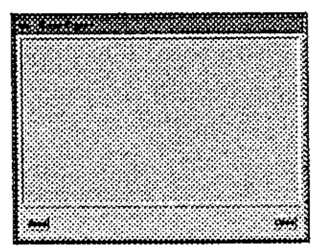
Plot window tools



Help



Bugs report



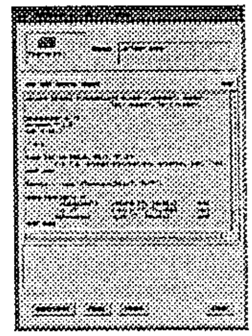
Metview/ws macro language

- Perform data manipulation and plotting from within the Metview/ws system environment
- Object-oriented design
- Example

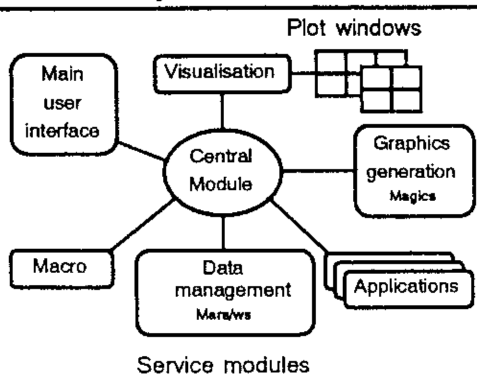
```
w = plotwindows ("height", 1000, "width", 1000)
r = retrieve ("date", -1, "param", ["u", "v"])
p = pcont ("wind_arrow_colour", "red")
plot (w, r, p)
```

- Calling Fortran routines
- Automatic building of GUI (coming)

Macro editor



Metview/ws system overview



System architecture

- Metview/ws is modular
  - Modules execute and/or ask for *services*
  - Modules may run on different machines
  - Modules exchange requests in a Mars-like language
  - Configuration is defined by user resource file
- User interface is independent of *service modules*
- Batch mode

Basic modules
Central module

- System configuration management
- Message switching
- Modules start-up
- Command recording

Basic modules
Main user interface module

- Icon management
  - Create, delete etc.
- Editing of definitions
- Drag and drop
- Initiates the *execute* action
- Displays messages from services
- Keeps the state of icons

Basic modules
Visualisation & Graphics generation

- Based on Magics
- Plot area management
- Geographic area and projection definition
- Zoom
- Animation
- Print (PostScript)

Basic modules
Data management module

- Based on Mars/ws
- Optimised data retrieval
- Formulae evaluation
- *Examine* action

Basic modules
Macro processing module

- Command language interpreter
- Commands are executed sequentially
- Executes more than one macro at same time

Application modules

- Generation of specific products
- No user interface
- Generally three phases:
  - Data request
  - Data manipulation
  - Visualisation
- Application modules in Metview/ws 1.1
  - Cross section, Average, Profile
  - Relative humidity, Coloured wind
  - Other application modules

Software tools

- X Window System
- OSF Motif
- X-Designer GUI builder
- C++, C and Fortran 77
- Xelion S-GKS (ISO GKS with X interface)
- Magics (ECMWF graphics package)
  - Conicon
- Mars (ECMWF)

The METVIEW/ws project

- Cooperative project between ECMWF and INPE/CPTEC
- ECMWF is also assisted by a staff member from Météo France

Plans

- Application modules
  - Metgram
  - Porting of 30+ Metview/batch applications
- Full support for title and legends
- Colour image editor
- Improved support for observations
- Batch version
- Macro enhancements

Summary

- Version 1.1 has been released at ECMWF
- Modular and expandable system
- Icon based user interface
  - Create definitions
  - Actions
- Mars data retrieval
- Magics visualization

Deutscher Wetterdienst

Germany

MAP - The Meteorological Application and Presentation system of Deutscher Wetterdienst

by D. Polte, M. Pogoda (DWD)

With MAP Deutscher Wetterdienst is developing a software system for weather forcasters with a Graphical User Interface (GUI).
This MAP system bases on the well known standards:

UNIX and C,
X-Windows and OSF/Motif,
TCP/IP and ISO/OSI protocols,
GKS.

The main aim of the system is to combine various data on the screen in an easy and fast way.

1. Experiences with GUI

Developing the presentation software startet in 1988 with personal computers.
For DPSU, the Data Presentation System (UNIX version), running both on PC and on workstation, we used an alfanumerical interface based on "curses" and buttons constructed with the help of GKS.
To control with "curses" the keyboard had to be used because of a big loss of performance with a mouse. The number of possible buttons was restricted with problems to arrange them on the screen.

To avoid all these problems lead to the decision to use X-Windows/OSF Motif for the development of MAP.
There were first results with X-Windows/OSF Motif in 1992 with the main menue and

some other parts of the software for SWIS, the Road and Weather Information System of DWD.
In the end of 1993 the XDesigner was acquired for the further software developement. The decision to acquire it was highly influenced by EGOWS reports.

Supported by the experiences with XWindows/OSF Motif the XDesigner could be used immediately. There were soon first results.
The MAP layout was now fully developed using the XDesigner. As expected every change is easy to make.

Within the software developer team there is one design specialist for the layout who coordinates the programming of the various callbacks, wich will be done by the other experts.
Significant for the layout were survey, functionality, quick choice.

The first real test of MAP took place at the International Aerospace Exhibition (ILA) fair in Berlin-Schönefeld. Five meteorologists of the Potsdam Weather Office were instructed by the software developers at the fair in a short time to use the system with its new layout. They then succesfully presented MAP to the visitors.

2. The MAP-Interface, a description

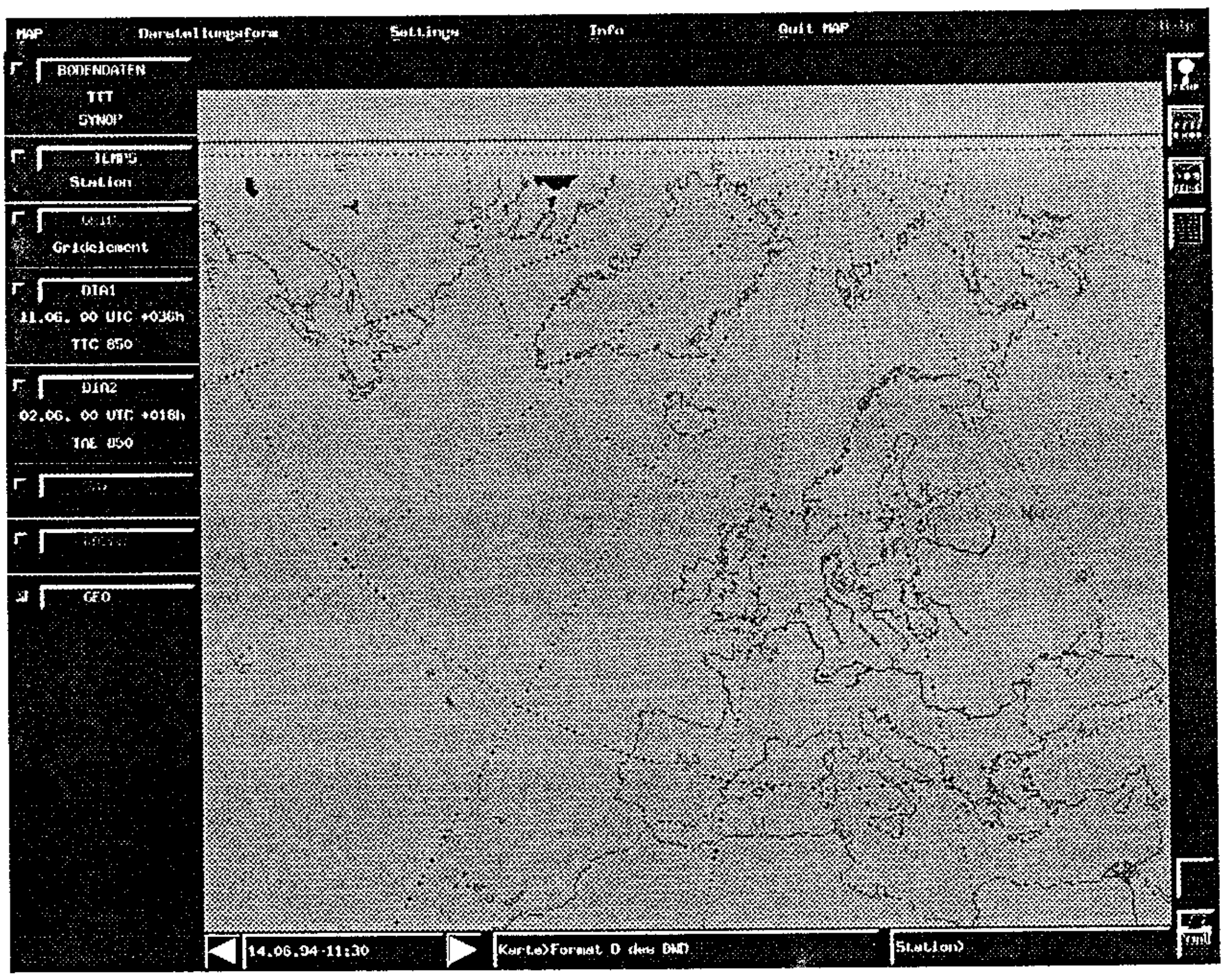
Every interactive module follows the general rules of layout described here.
The three basic parts of the window layout are

a menue bar at the top of the window,
a status bar at the bottom of the window and
a working area in between.

The main module will be the most complex presentation module. So there are much possibilities to present and manipulate data. Therefore, it is necessary to implement different menutopics, pushbuttons, short cuts a.s.o, widgets to control all this functions

of the software module.
We decided to have only a second level of choice to have short cuts.
So we put control areas to the left and right side of the working area.
In the main module the working area is a drawing area widget based of a GKS (GTSGRAL-GKS X11-driver as subwindow).

A second menue is placed between the main menue and the drawing area. It is a context sensitive menue, depending on the displayed data.



The following description will go more into detail:

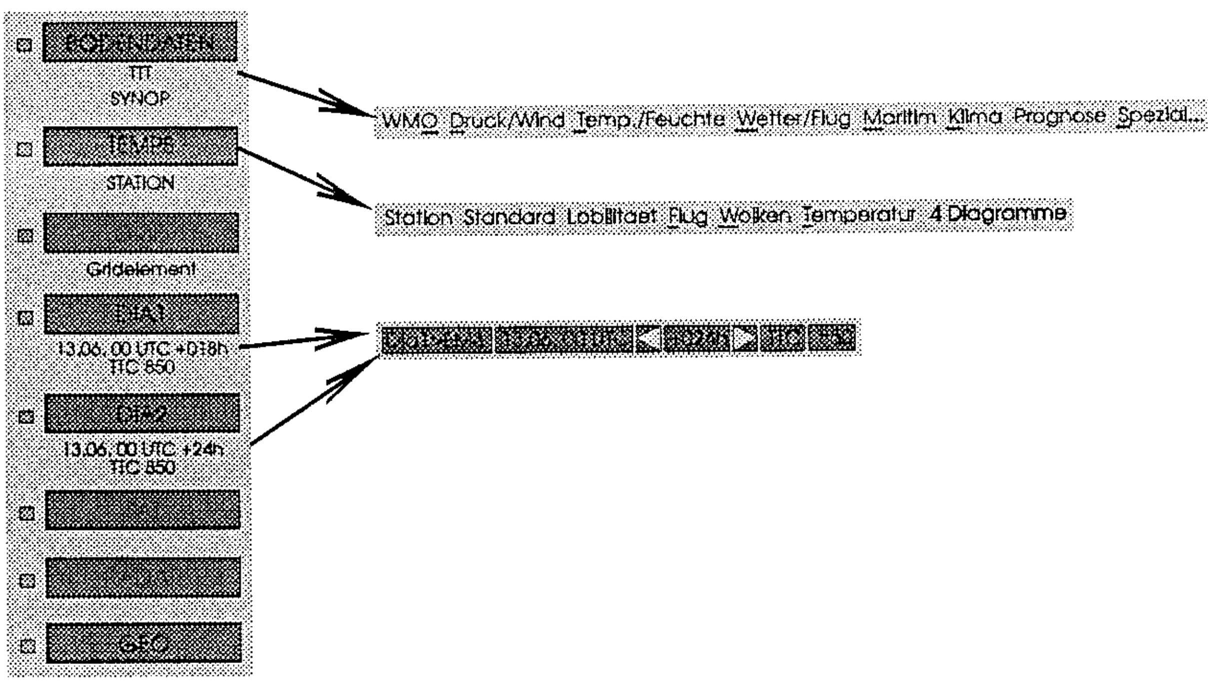
The main menu includes some topics to contol the application on a top level. It consists of

- MAP,
  integrating the "Interaktives Graphisches System" (IGS) and the satellite software. There will be also a possibility to archive data of interesting weathers.
- Darstellungsform
  enables to switch between different representations of data like digits, symbols or color points.
- Settings
  to vary system parameters, e.g. which data server or which colormap will be used.
- Info
  to turn on or off a legend for each data presented on the screen.
- Quit
  to exit the application.
- Help
  to start up a help system, which isn't implemented yet.

The status bar at the bottom of the window shows the selected termin, map and station, which can be changed by clicking the pushbuttons selecting from a list.
Depending on the actual time observational or prognostical data are chosen.

On the left is a panel to control the display of different data.
Each data type is represented by a radiobutton, a pushbutton and a label. The pushbutton shows the data type and brings up a menue, which will be desribed below.

With the radiobutton the data representation can be included or excluded. The label displayes the type of data representation.
At the moment it is possible to select observations and direct model output (ground and temp) and model output as metafile.
The geographical background is handled like a normal data type. So it can switched on or off.
From this panel it is possible to overlay at one time as much data types as the modul can handle at all. This would not be the best way get all information in a short time, of



course.

With new data types or functions the amount buttons will grow.
The pushbuttons beside the radiobuttons bring up the different context sensitive menues under the main menue. With these menus the form displaying the data type can be changed. This mean that it is possible to change between different types of display types like the standard WMO synop model or others.
For special data types like modell output as metafile the metafile can be selected by

selecting: model, model run, valid time, element and level.

On the right there are buttons for

temp evaluation,

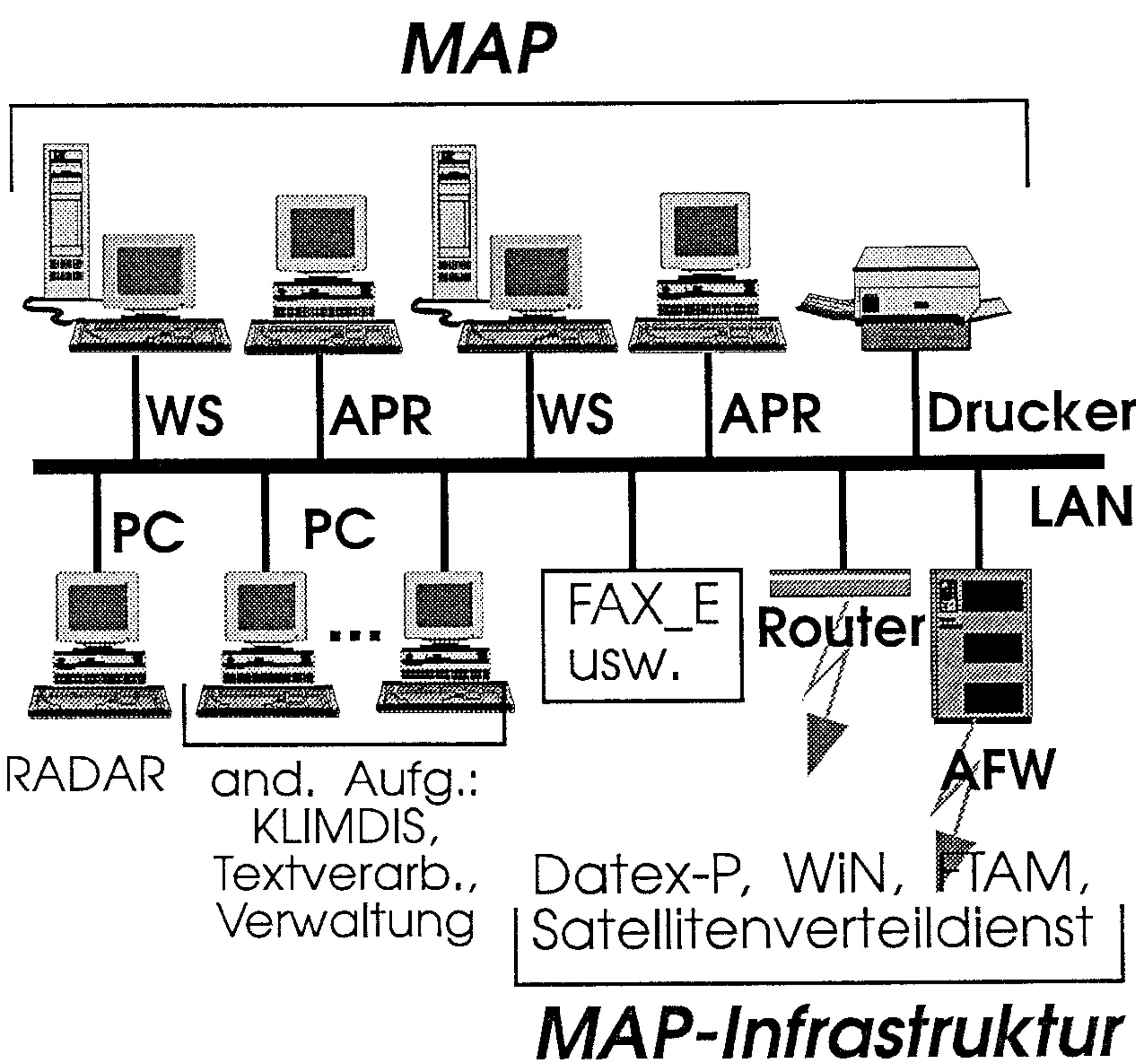
meteogram display from DWD's Europamodell,

graphical representation of time series,

alfanumerical presentations of synops (time series, extreme values a.o.),

zooming (not yet),

printing.



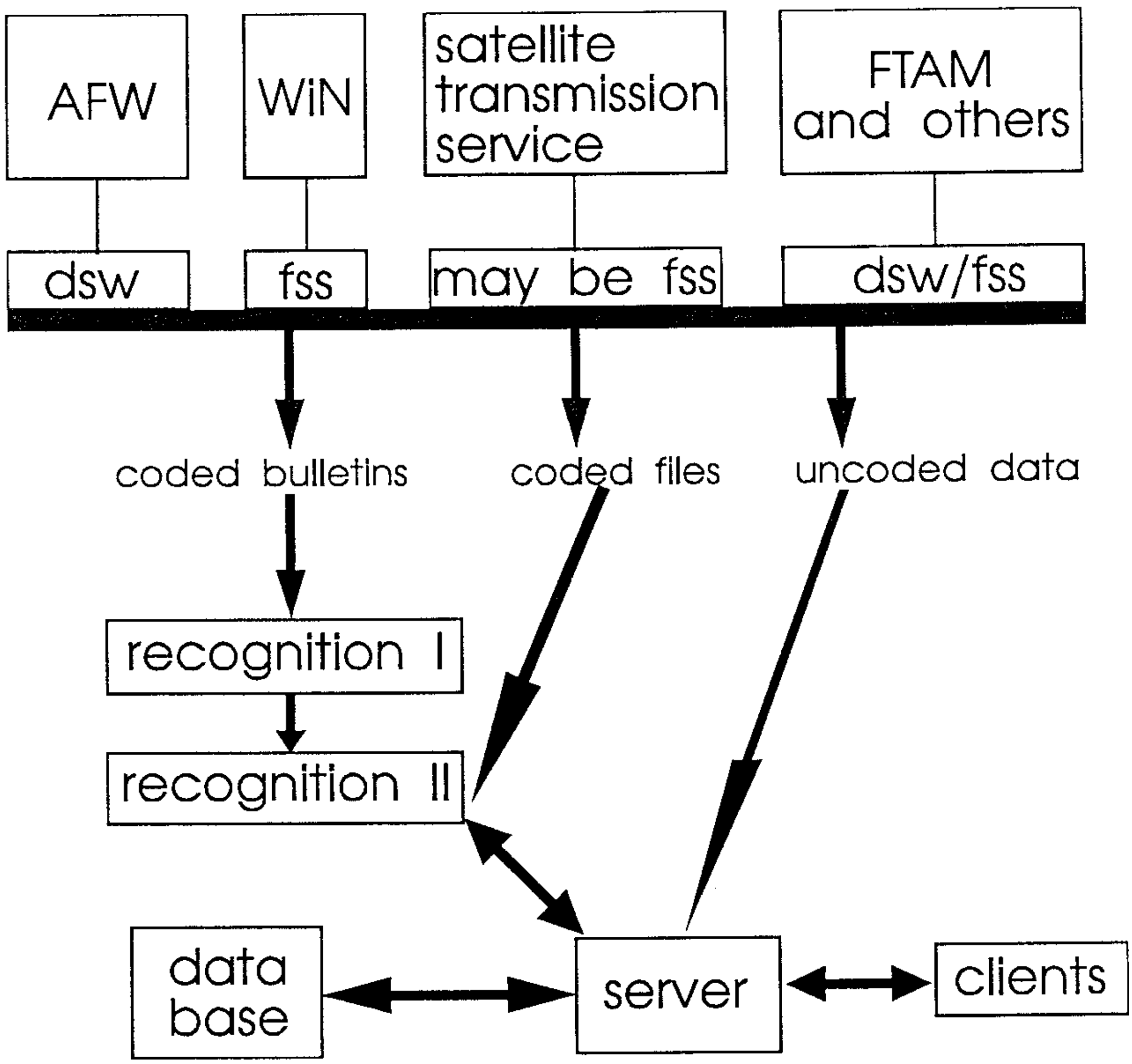
3. Data storage and communication

It was not possible to handle all data with one server. So we had to install a second server.
The first server DSP (DataServerProcess) handles bulletin data.
It has high transfer rate because of the quasi continuos data input. For this server it is typical that most data have a fixed record length. Only grided data (FM47) don't follow the rule.
The second server MFS (MetaFileServer) handels data without a fixed record length and large binary objects.
We had to put the network part of the software on a more abstract level.
So on UNIX System V Release 4 the transport layer interface (TLI) is used. Using TLI allows to handle different connection oriented transports.
At the moment there is only TCP available on our platforms. So we have no experience in the use of different protocols.
The new TLI based communication software can operate with our older socket based interface.

The local area network (LAN) and the entries to the wide area network are shown at the top of this chapter.
The different connections have different protocols and underlying software. Two software packages control the different protocols.
On the WiN (Wissenschaftsnetz) the transport of data (metafiles, satellite images and other) is done based on ftp.
The control of the ftp transfers is done by the FileSwitchingSystem (FSS).
So it is possible to give the files priorities. It adresses the files to users into their directories or to the MetaFileServer.
The FSS can also handle transfers done by the OSI FTAM.

Asynchrone lines, fax cards and other devices are controled by the DSW package (DataServiceWeatheroffice).

DSW is able to select data from the data base, to put them together with texts and to transmit this product to a consumer of weather services by a time table.



Procedures that produces graphical output automatically will be implemented int the second half of the year.

In August 1995 the old facsimily transmission will be replaced by satellite.

A test will start at June. FAX_E will be available from August 1st. After a short time of doing both transmissions the old service will be canceled.
From this time the wide area transmission of files will be done by this new service. The FSS will be the terrestrian backup system for this FAX_E.

4. SWIS

The Road and Weather Information System (SWIS) will be integrated into the MAP system.
That means, that all data of MAP and SWIS data can be shown together.
SWIS activities will be done all over the south of Germany.

5. Warn control and nowcasting

Checking for dangerous weather elements shall be done for each synop or metar of a defined area. The meteorologist will see this dangerous weather element in a special window, a warn monitor. This will be the first step to a better automatical weather watch.
It will lead to a nowcasting system.
DWD and the Freie Universität Berlin will develop a system of nowcasting procedures that operates automatically. The MAP database including all observational data, model output, radar and satellite images gives the nowcasting procedures their input.
The output of the nowcasting modules will be displayed in the MAP system, of curse.
The work for this project started in spring 1994.

6. Problems and Workarounds

At the moment there are only parts of the MAP system implemented with a OSF/Motif based GUI. Integration of existing software parts with non-X-Window and non-OSF/Motif interface means redesigning these parts to build a homogenous system.

Displaying images within a GKS context. It shall be tryed to do this with a pixmap and a nonstandard GKS function.

The old problem of GKS's limited performance still exists.

A more efficient way is needed to transmit data from the message switching system to the server including the decoding. At this time their are to many transfers to and from the disk.

In 1994 we will replace all PC's for presentation purposes by SGI Indys.

There will be better separation between communication and presentation software.

U.K. Met. Office

United Kingdom

The Practical Development of Graphical User Interfaces in the Horace Project

R Townsend and P Trevelyan
U.K. Met Office
Bracknell U.K.

1) Background

From the start of the Horace Project it was decided that Graphical User Interfaces (GUIs) would be created in Motif. It was also apparent that Motif has some similarities to Microsoft Windows which our users are more likely to be familiar with. It was also considered possible that we may buy in some commercial software packages to use in Horace and these would most probably use Motif as well. It was fairly obvious that writing Motif programs from scratch would be an enormous task, so a survey of GUI building tools was undertaken. As a result of this survey **X Designer** from Imperial Software Technology was selected and two seats were purchased (later increased to three seats).

As the Met Office has a large body of existing application software and expertise in Fortran and GKS, it was decided that the actual applications should be written in these languages. It was also decided that the GUIs that were to control the applications would exist as seperate Unix processes. The advantages of this would be that either the GUI or the application could be modified independantly, as long as the interface between the two remained the same. The disadvantages are that extra software would have to be incorporated to facilitate communication between the processes as well as handling the unexpected death of either of the processes. When the user selects an option from the Horace Main Menu the GUI is started first. It is the GUI itself that starts the Fortran application. The GUI and the application communicate with each other using Unix message queues and files.

2) GUI Development Process

When a member of the GUI Team is assigned the task of creating the GUI for a particular application, their first step is to consult the **Functional Requirements** that specify the functions and selection options that are needed. If there are any existing applications on the **Oustation Display System** (ODS) or on the **COSMOS** (HDS mainframe system) terminals in the CFO (Central Forecast Office), the GUI Developer will examine these to gain a deeper understanding of how the application currently operates. In parallel with this activity an Application Developer will be looking at the same problem from the point of view of the Fortran & GKS program.

Once the two Developers have a reasonable understanding of the problem, a meeting is held at which it is decided what pieces of information need to be passed between the GUI and the application. The new GUI and application processes are then added to the **Task Model** (see fig1) that is maintained using the Software Through Pictures (STP) C.A.S.E. tool Data Flow Editor. The definitions of the messages to be passed on the Unix message queues and the intermediate files are entered using the StP Data Struture Diagram editor (see fig2).

Because it is relatively easy to create prototypes using X Designer the GUI Developer is able to quickly produce several inital versions. These are constructed following the Horace GUI Team Standards (see appendix) which are based on the Motif standard and are intended to give a consistency to all the GUIs within the Project. At this stage representatives from the users are invited to see the prototypes and much useful feedback is obtained which helps refine the prototypes so they more closely match the users' requirements.

Once the users have accepted the layout of one of the prototypes, the functions that will be executed by the GUI are designed and written. It is usual at this stage to connect the GUI up to a simple test application that can simulate the interprocess communication.

When the GUI is working correctly it is connected to the real application and the two processes are tested together, either using captured non-real time data or actual data from the Horace Database. Testing involves not only ensuring that the expected functionality works but that the output of the application is accurate. If amendments are required then the Task Model and the Data Structure Diagrams are reviewed to ensure they are up to date.

After the new GUI and application have sucessfully passed their testing phase they are transferred to the HORACE"next_release" directories so they can be given a complete integration test with the complete Horace system. Only after passing the integration test is the new software transferred to the operational system and installed at the users's sites.

In the early days of the Horace Project when there was little experience of creating GUIs a certain amount of experimentation took place. This resulted in a lack of standardisation between different GUIs. It was also recognised that we had not made enough use of things such as pulldown menus, accelerators and mnemonics. To remedy these problems we have now developed a set of standards that all new GUIs must adhere to and we are also upgrading our existing GUIs to meet those standards. (see appendix)

One policy that we have adopted is not to hard code colours into our GUIs but to have them use the colours set by Hewlett Packard's Visual User Environment (VUE). In the next major release of Horace our Main Menu will be replaced by a customised VUE Front Panel. This will give the users access to Style Manager which will allow the user to adjust fonts etc. to their own preferences, which is of particular importance to people who are colour blind.

FIG 1

ipc: level 0

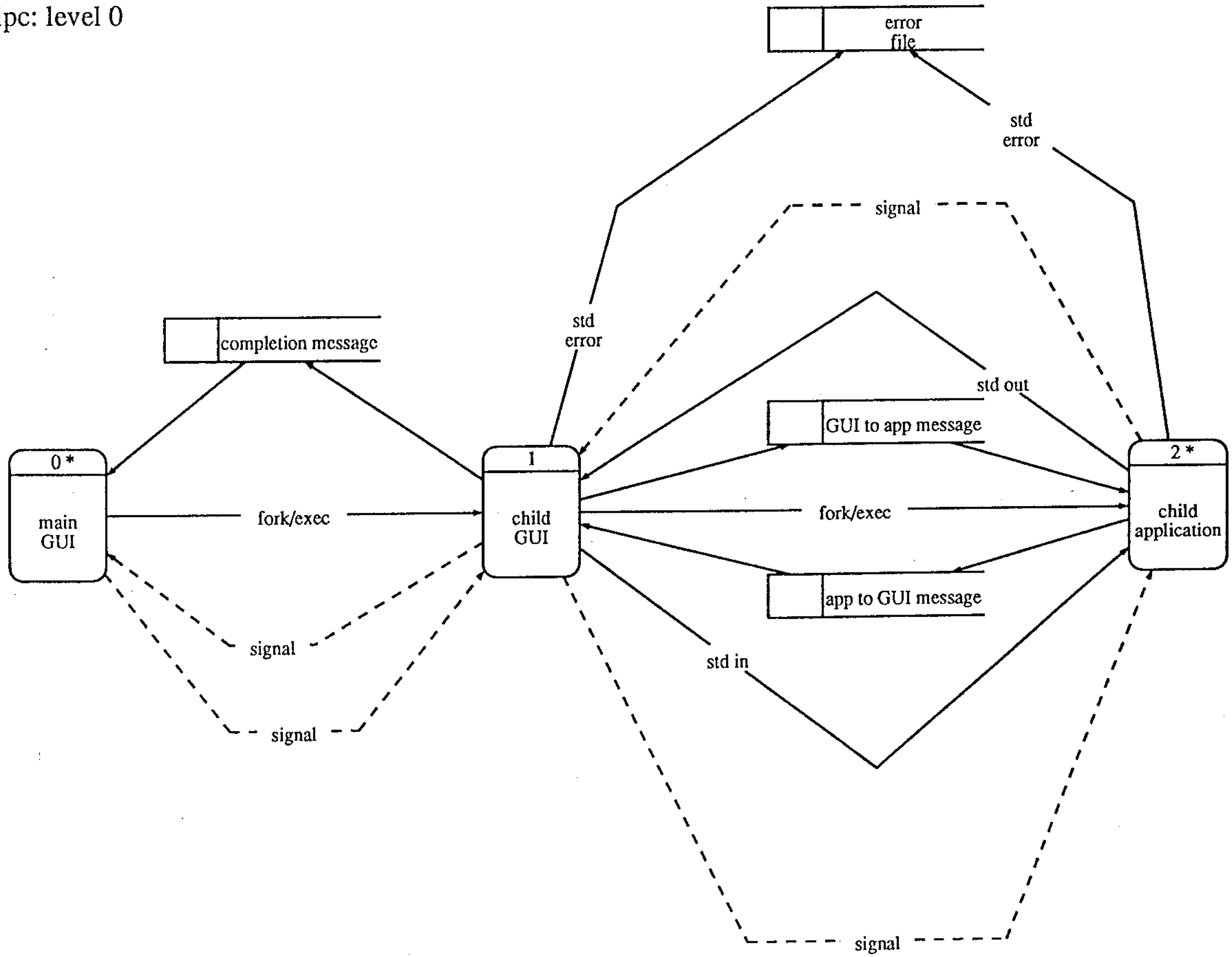
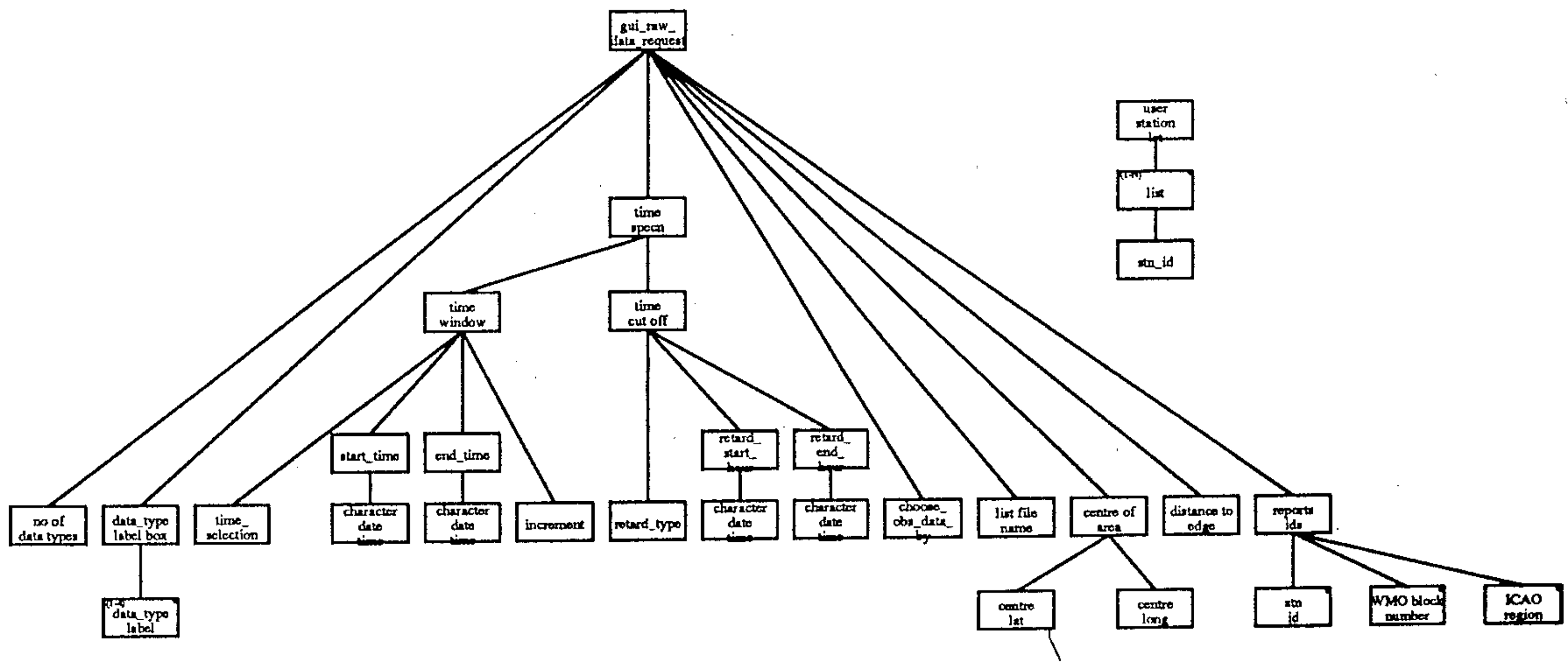


FIG 2

gui_raw_synop_request



WMO

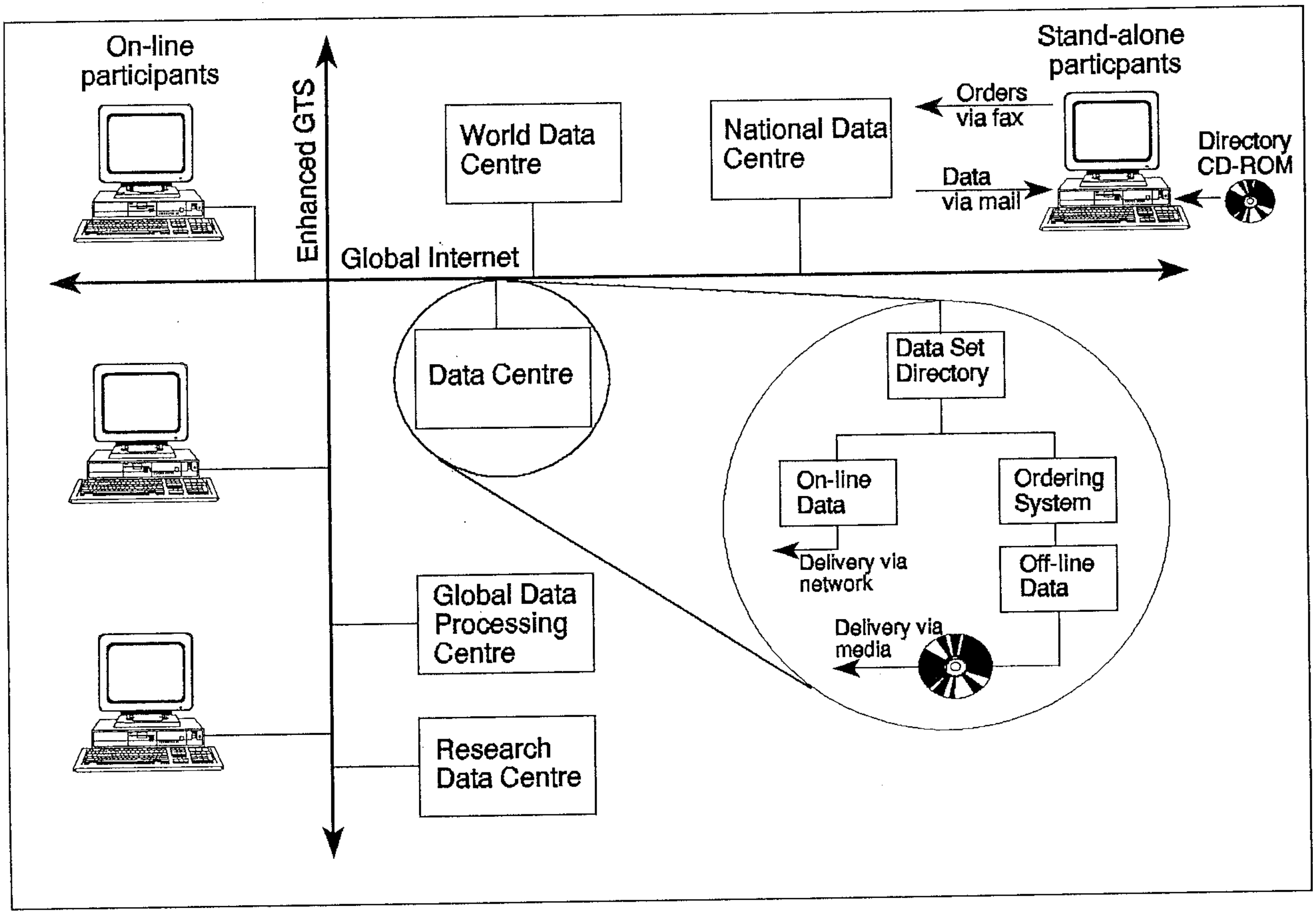
WMO Distributed Databases (DDBs) Project

Purpose

Provide data and information needed by WMO, and related international, programmes but not routinely exchanged on the GTS.

Vision

International set of distributed databases accessible as if a single data base. Would provide access to on-line and off-line data for transport via networks and mailed-media.


On-line participants
Enhanced GTS
Global Internet
World Data Centre
National Data Centre
Stand-alone particpants
Orders via fax
Data via mail
Directory CD-ROM
Data Centre
Data Set Directory
On-line Data
Ordering System
Off-line Data
Delivery via network
Delivery via media
Global Data Processing Centre
Research Data Centre

Responsibilities (WMO CBS)

Working Group on Data Management

- Develop a logical data model so the other CBS Working Groups can understand the requirements placed upon the Systems they manage, and proceed to design and implement the systems required

- Develop the request/reply mechanism and necessary standards

Working Group on Data Processing

- Encourage rapid implementation and use of file servers providing FTP access to meteorological data sets

- Develop and implement DDBs servers

- Ensure the DDBs clients are implemented to provide an interface between the users and the servers (in cooperation with other interested groups)

Working Group on Telecommunications

- Offer Internet-like services on the GTS Main Telecommunications Network within 5 years

Guidelines

- The DDBs should be globally coordinated and serve all WMO Members.

- The DDBs should provide a standard set of functions including the identification of the:
  - existence,
  - location(s),
  - accessibility and
  - delivery

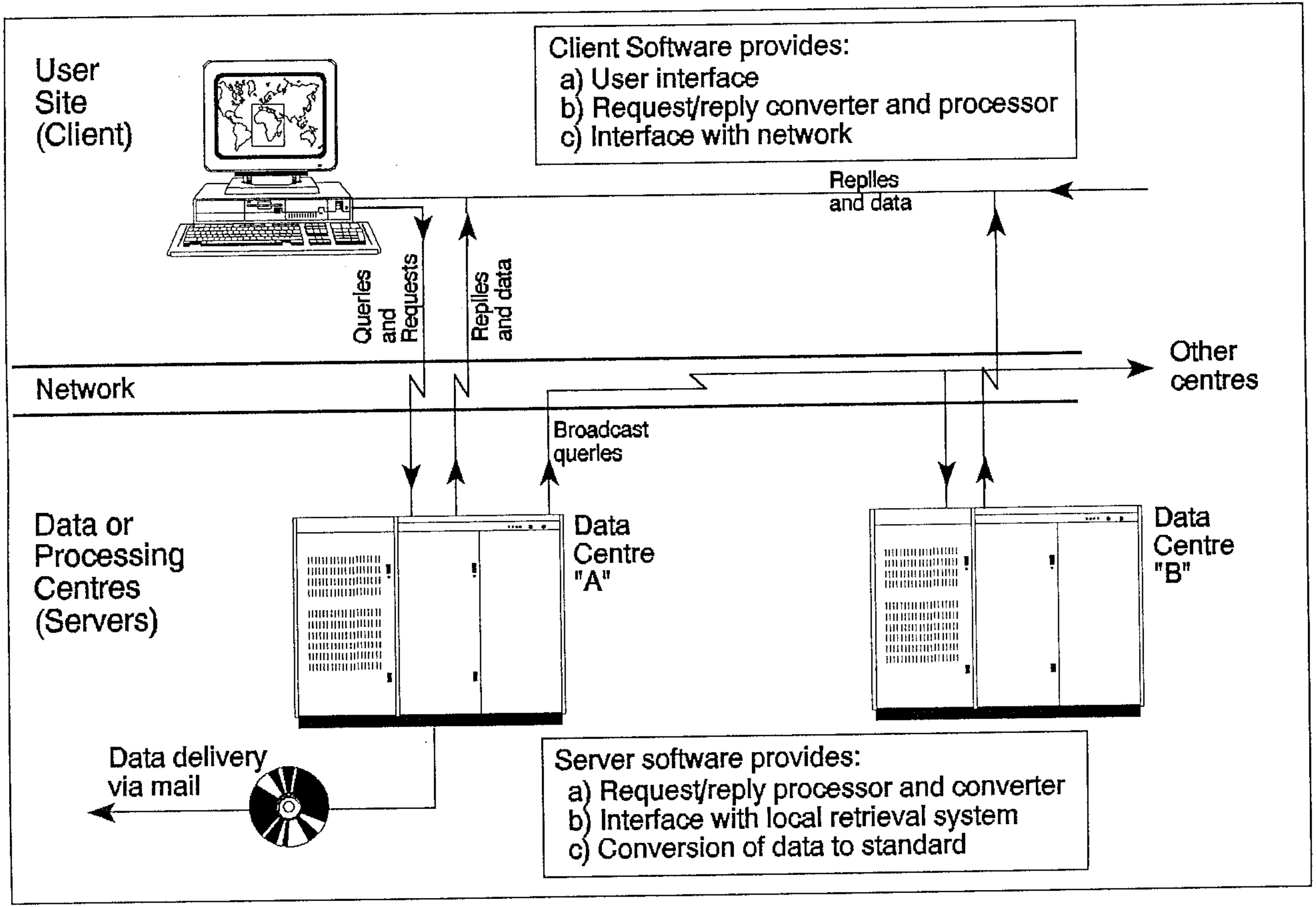
  of the data.

- The DDBs should conform to a set of implementation standards including:
  - request/reply mechanisms,
  - request/reply content, and
  - transmission mechanism(s),
  - data representation.

- A method of ensuring DDBs conform to the above standards should be developed.

Approach

Client-server paradigm


User
Site
(Client)
Client Software provides:
a) User interface
b) Request/reply converter and processor
c) Interface with network
Replies
and data
Queries
and
Requests
Replies
and data
Network
Other
centres
Broadcast
queries
Data or
Processing
Centres
(Servers)
Data
Centre
"A"
Data
Centre
"B"
Data delivery
via mail
Server software provides:
a) Request/reply processor and converter
b) Interface with local retrieval system
c) Conversion of data to standard

Goals

Two-years

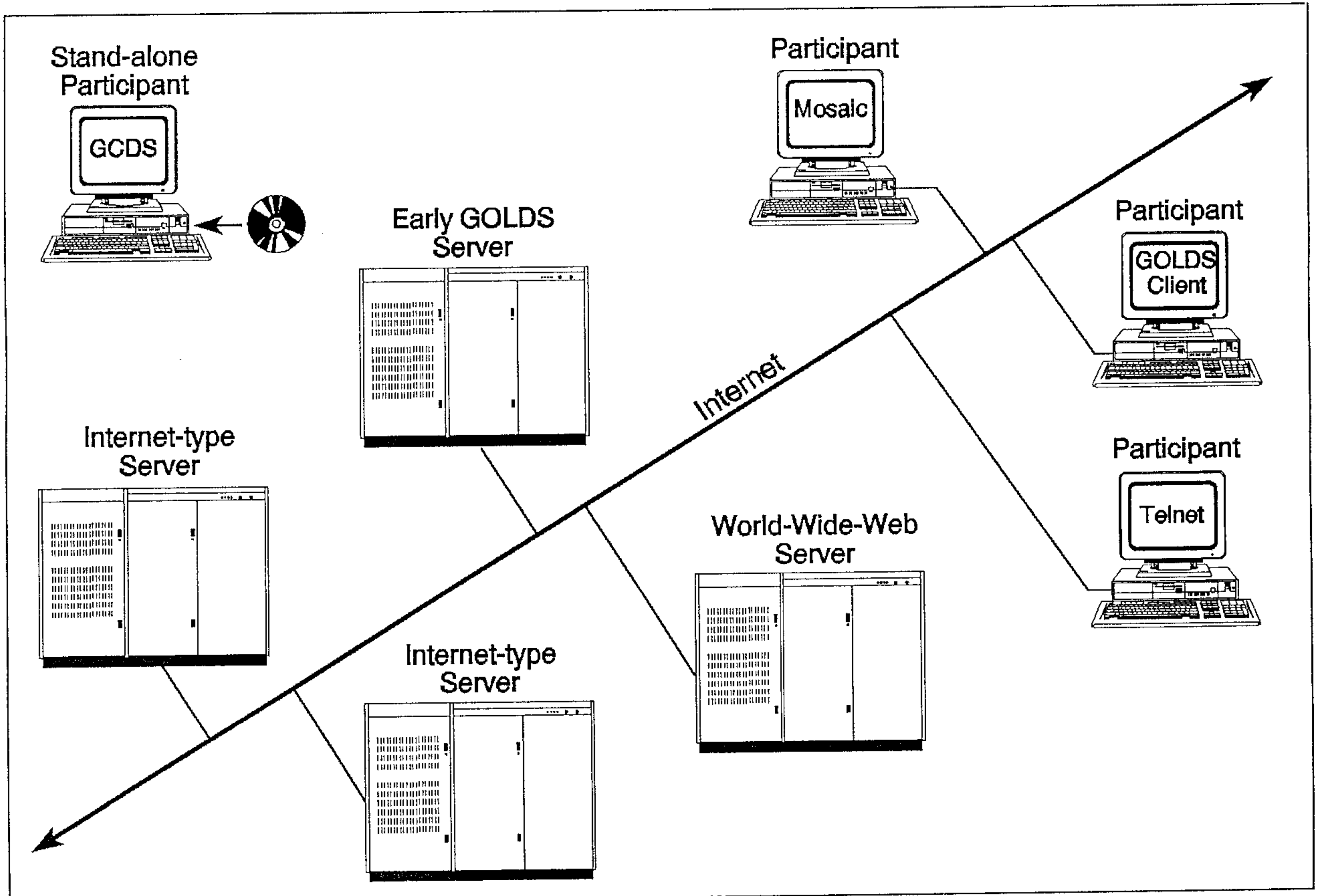
Series of FTP-type servers, following standard naming conventions, offering meteorological data accessible over the Internet

Five-years

Series of linked distributed data bases implemented as a client-server process. High level directory information would be exchanged between centres so users could identify data available from all centres by accessing a single site. If users wanted to retrieve data from a remote site, they would interact with that site directly.

Long term

Series of on-line systems accessible as if a single data base. Each DDBs centre would maintain its own data catalogue and ordering system which would be able to interact with client software running on user machines. Client queries could be directed to a known site or could be broadcast to all server sites.


Stand-alone
Participant
GCDS
Early GOLDS
Server
Participant
Mosaic
Participant
GOLDS
Client
Internet
Internet-type
Server
Participant
Telnet
World-Wide-Web
Server
Internet-type
Server

Required Standards

Description of the contents of DDBs

Representation of DDBs requests

DDBs requests:

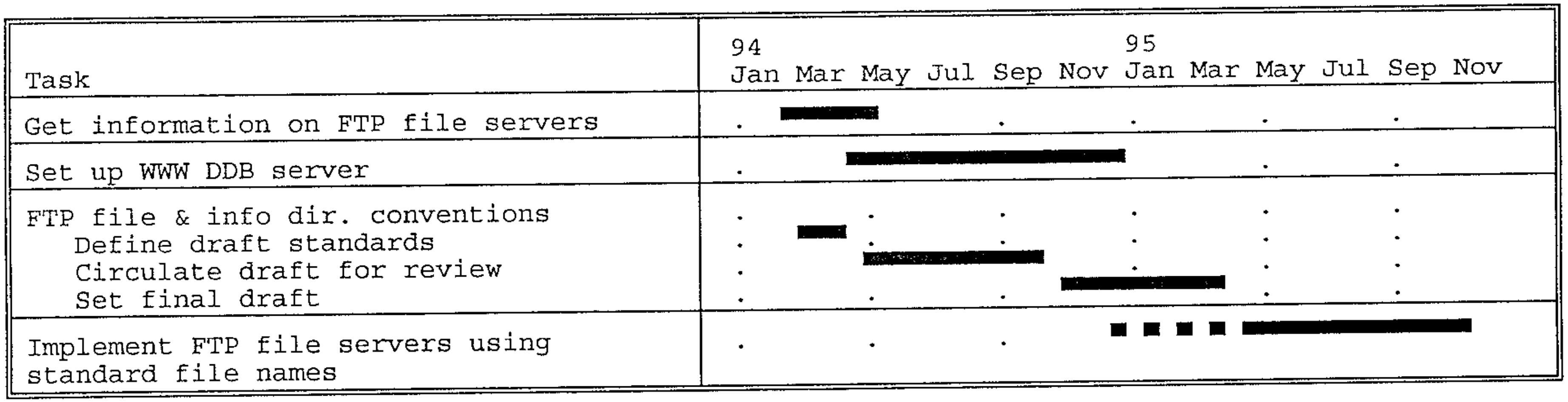
a) Queries (do you have...)
b) Requests (provide a copy of...)

Representation of DDBs replies in the form of

a) an acknowledgement
b) an action to be taken
c) the requested data or metadata

Initial Trial Implementation

FTP File Servers Accessible via Internet



Longer Term Activities

Prototype Client-Server Systems

Task	94 Jan	95 Jan	96 Jan	97 Jan	98 Jan	99 Jan
Request/reply representation stndrds						
Develop draft client-server design including req/rep stndrds						
Collect draft design and standard proposals						
Circulate drafts for review						
Set final draft standard						
Develop data/metadata representation standards						
Trial DDBs implementation						
Final WMO-approved standards						
Review trial and develop final						
Obtain approval of WGDM						
Obtain CBS approval						
Operational client-server DDBs						

Zentralanstalt für Meteorologie
und Geodynamik

Austria

Central Institute for Meteorology and Geodynamics, Vienna, Austria

Ing. M. Göstl

TPVIS

TAWES Project Visualisation

Hardware:

SUN SPARCstations xx with SUNOS 4.1.x
Laser-, inkjet- and thermotransfer-printers, electrostatic plotters (A0 format)

Strategies:

Usage of standards:

UNIX, C, X11, GKS, CGM, (XViev)

Usage of existing Programs:

MAGICS, Trajectory calculation, CALMET, ...

Functions Summary:

Superimposition of all available meteorological informations
Single parameter display in pseudo colors
7 zoom/pan functions
Macros (saving of selections)
Plot functions

Display:

Observations: TAWES, Synop, Metar, Environmental Observations, Temps
as WMO station model, free station model (free selection of parameters) or timeseries

ECMWF-forecasts (more than 20 parameters)

Analysis charts

Trajectories from main cities and atomic power plants aswell as trajectories calculated on request

Satellite images: IR, VIS, WV

New features compared to TPVIS presented at EGOWS 4:

Metar observations displayed as station model
Combination of IR and VIS satellite images
Cutline and time series extraction from satellite images
Display of windfields of small regions (CALMET project)

Future plans for 1994:

Radar images
Lightning Observations
Front Editor
Batch Plotting
(Animations ?)

Migrations:

SUNGKS	to	Xelion S-GKS:	Summer 1994
SUNOS 4.1.x	to	SOLARIS 2.x:	End of 1994
OpenLook	to	CDE :	1995 or later

Danmarks Meteorologiske Institut

Denmark

Semi-automatic sea area forecasts

Thomas Madsen, Danish Meteorological Institute

1.0 Introduction

The DMI has the important duty task of issuing wind and visibility forecasts for the waters surrounding Denmark, Greenland and the Faroe Islands. As this has to be done several times each day, it is one of the most time consuming activities of the forecasters.

A system for semi-automatic generation of sea area forecasts has therefore been developed by a group at the DMI. The system converts the prediction of the weather models into words, which in turn can be visualised on the workstation. After being checked and perhaps modified by the forecaster, the forecasts should be transmitted to a telephone service and may be presented on the forecaster's PC for further editing. So far only raw model output has been used, and only 24-hour forecasts are issued. The system will only produce a prediction of wind speed and direction, and thus leaves visibility to the forecaster.

The system is divided into two separate parts:

- The MO-program (MO for Model Output) generates wind-forecasts for each of the 23 sea areas based on model predicted 10-meter winds. Forecasts are produced in the form of a code which may be converted into words or wind symbols.
- The HI-program (HI for Human Intervention) is a graphical user interface program which allows the forecaster to modify the wind forecasts and add weather and visiblility information by manipulating wind and weather symbols on the screen. Parts of the MO-program are used for generating new forecasts based on the alterations of wind symbols.

2.0 The set up

The MO-program and the HI-program are installed on different workstations.

- The computation of the automatic forecasts is performed by a workstation set up as a compute server (tera). 6 times (2 analysis * 3 models) every 24 hours a computation is initiated, leading to forecasts beginning every three hours for as long as there are projections available. This means that a computation of DKV forecasts at 2.40 UTC will lead to forecasts beginning at 3 UTC, 6 UTC, 9 UTC, 12 UTC and 15 UTC (We have projections of the DKV-HIRLAM until +36 hours). Of these only 3, 6, 9 and 15 will be used at present.
- The HI-program however is installed on various application servers, serving the terminals of the operations office. All the installations of the HI utilises the computations of tera, and they all save the (possibly) altered forecasts on tera (using terminal dependent filenames).

- For further editing the forecast is transmitted from the operation server to a VAX, from where the forecaster can retrieve the document to his PC for editing in Word-Perfect.

3.0 The MO-program

3.1 The set of permitted forecasts

For generation of forecasts a set of permitted forecasts has been defined. The definition of these forecasts were based on a study of previous forecasts made by the forecasters. Certain words used by the forecasters such as "later" and "or" has been discarded, as they are too vague to be given a clear interpretation. Each forecast has to be uncontroversial, meaning that each expression in the forecast can be interpreted as a single one wind direction interval and speed interval. The removal of vague terms is also in correspondence with a policy of the DMI to avoid vagueness in the forecasts.

Each forecast is expressed in the form of a code, as a string of characters. The string contains a series of tokens for wind force, wind direction, temporal changes and geographical subdivisions. Forecasts having the same token types in the same order belong to the same *pattern*. The system contains 131 such patterns, the simplest one being 'D, F, .' or 'wind Direction, wind Force'.

There are 16 possible directions each with an interval of 45 degrees, which means that a prediction of west means between westsouthwest and westnorthwest. For wind speeds below 15 m/s the possibility of predicting a 90 degrees interval for 8 directions exist. For wind speeds below 5 m/s variable wind is also allowed. Wind force is predicted in partly overlapping intervals: 0-5 m/s, 3-8 m/s, 5-10 m/s and so on. Wind can change over periods of 3 hours (T), 6 hours (t) or for direction only: 6 through 24 hours (Dt).

Several limitations has been inposed on the choice of wind changes and differences. An example is that wind direction has to veer/back at least 45 degrees for the system to accept the wind change. This means that a forecaster, that isn't familiar with the limitations will see his modifications of the forecasts further modified. These limitations has been introduced in order to keep the forecasts simple and understandable, when possible.

The number of permitted forecasts resulting from this set of rules and limitations totals $5*10^{11}$. Although very much smaller than the (in principle unlimited) number of forecasts available to the traditionally working forecaster, this is still a huge number to choose from.

3.2 Choosing the right forecast

As we have now defined a table containing $5*10^{11}$ forecasts, the next problem is to choose the one best fitting the situation as predicted by the model. We use the 10-meter wind in 3-hourly projections as the basis for the selection. To make the comparison we interpret the forecasts as a probability forecast of wind direction and force. For every time projection and, if a geographical subdivision is present in the forecast, for each of the relevant sub-areas, the wind prediction defines a two-dimensional distribution of the probability of encountering different values of wind force and direction. Each grid point wind is taken as an instantiation of the truth that we attempt to describe. All the grid point values taken together form the whole truth.

The degree of match is measured by a Brier score (Murphy, 1985), in our system taking the form

$$BS = \frac{1}{TGDF} \sum_{t=1}^{T} \sum_{g=1}^{G} \sum_{d=1}^{D} \sum_{f=1}^{F} \left((p_{d,f}(t,g) - \delta_{d,f}(t,g))^2 w_f \right)$$

Here T and G designate the number of time projections and grid points in the sea area, respectively; D and F are the number of possible values of wind direction (whole degrees) and speed (whole number of m/s). The function p is a prescribed probability distribution, dependent on the forecast being evaluated, giving the forecast probability that the wind at grid point g and time t has direction d and speed f. The δ_{df} is 1 if this in fact is the case, zero otherwise.

The w in the formula is a weight factor dependent on wind speed. It is equal to one for wind speeds below 10 m/s and increases exponentially for wind speeds above 10 m/s, with steps at the lower limits of gale and storm. This results in a better prediction of the highest wind speeds, and also leads to a better rate of detection of gale and storm.

Test runs with the DKV-HIRLAM model indicated that adding 5% to the wind speed of the model improved the verification results, which is why our truth is in fact the 10 meter wind times 1.05.

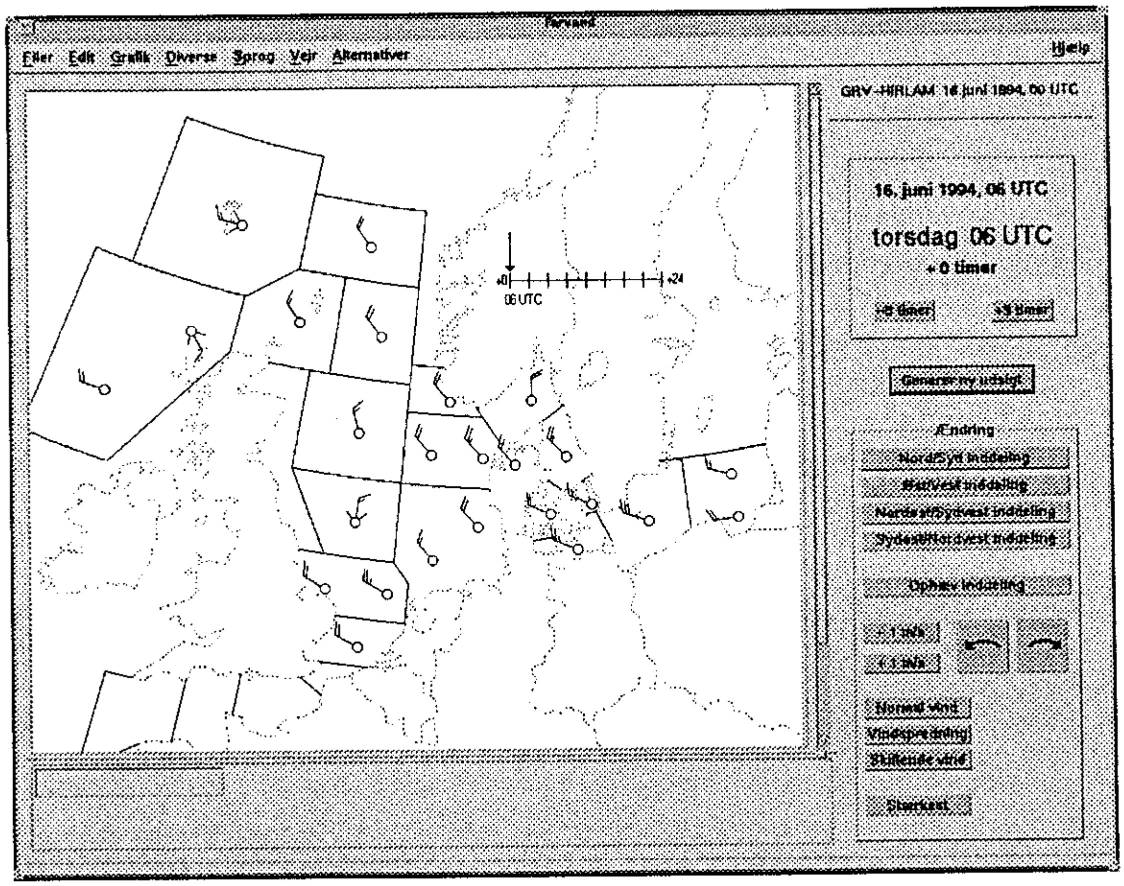
In order to reward simple forecasts, a complexity factor has been introduced. If two forecasts score approximately the same, the more simple of the forecasts will be chosen. This means that a complex forecast, in order to score better than a simple one, has to be *much* better than the simple one.

Not all $5*10^{11}$ forecasts are being verified during a computation, as this will require much more CPU-time than is available. Instead we classify the forecasts into 20 classes. The 5 classes best fitting the actual situation is then chosen and searched for good forecasts. This leads to a computation time for 23 sea areas of between 5 and 15 minutes, depending on the situation. Complicated wind fields containing fx cyclones will generally increase the computing time.

4.0 The HI-program

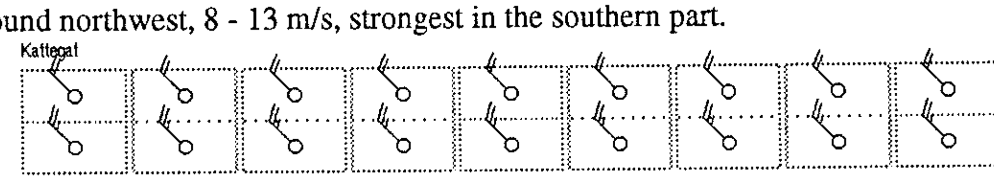
Using the HI-program a forecaster can modify the automatically computed wind forecasts, and add predictions of visibility and significant weather phenomena. The set of allowed forecasts is the same as before with the inclusion of the weather and visibility forecasts. This means that the forecaster is not entirely at liberty in the choice of phrases and complexity. He has to comply with the predefined set of forecasts or else the HI-program will alter his corrections to match an allowed forecast. This may seem as a severe handicap for the forecaster, but given time and practice the forecaster will hopefully find, that the set of forecasts can handle nearly all possible wind evolutions.

Upon start up of the program, and after model and time has been selected, the screen could look like this:



In this mode (the overview) the forecaster can time step through the forecasts, and thus see how the wind evolves in the span of the 24 hours. He can also.select the sea area(s) he wishes to modify and edit these by changing to time-series mode. This will display the time evolution of the selected areas. In this situation a time-series mode selection of Kattegat will look like this:

Around northwest, 8 - 13 m/s, strongest in the southern part.

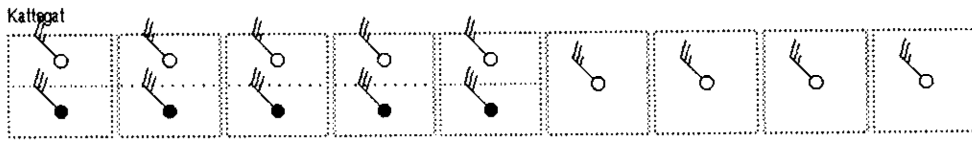


Notice that the speed of the symbols is the maximum speed of the forecast interval. That the wind is strongest in the southern part is indicated with a dotted line, and slightly lower wind speeds in the northern part. A full line would have indicated that the speed in the northern part was between 5 and 10 m/s, leading to this forecast:

Around northwest, 8 - 13 m/s, in the northern part 5 - 10 m/s.

As none of the four alternatives proved to be satisfactory, the forecaster in this situation chose to modify the wind field above in the manner shown below:

Around northwest 10 - 15 m/s, strongest in the southern part. From late this evening around northwest 8 - 13 m/s.



At this point the forecaster should insert the weather and visibility symbols. This is done by selecting phrases from a list of 41 possibilities, and inserting them in the relevant places. So far only two different weather symbols is allowed, that is to say, the weather can only change once during 24 hours.

5.0 Problems and plans

The MO-program has at present problems interpreting very low wind speeds. There is a tendency to overforecast weak winds, and the complexity of the forecasts is too high. With regard to the overforecasting, we believe the problem to be in the definition of the probability dsitribution for weak winds. As for the degree of complexity, an effort has been made to worsen the score of complex forecasts at low wind speeds, by increasing the complexity factor for wind speeds below 8 m/s.

As for the HI-program, it is a major problem that several of the forecasters very seldom gets a chance to use it. Lot's of practice is needed, as only a thorough knowledge of the set of permitted forecasts will avoid the inevitable frustration of having to see your modifications altered again and again.

In the near future, we will attempt to reduce the bias in wind speed by Kalman filtering or some other statistical interpretation technique. A critical point here is the difficulty in getting reliable and representative observations for the sea areas.

Wind forecasts could be forced to agree with observations at the beginning of the forecast period. Methods to check the quality and consistency of observations would be necessary.

A first guess of weather and visibility could be calculated.

Looking further out a system could be implemented, that enables the forecaster to rectify the isobars of the model, and thereby altering the wind field resulting in new forecasts.

Reference

Murphy, A.H.,1985: Proposed standard procedures for verification of local weather forecasts.

Workstations at the Danish Meteorological Institute

Knud E. Christensen

June 1994

Three main issues marked the past year: the new offices for the central met-office, the new SUN operating system and the transition from OpenWindows to Motif.

In developments we have so far been concerned with presentation of data, but now the first tool for generating the actual forecast text has been designed, a system for automatic generation of forecasts for sea areas.

New central met-office

The new office is up and running with the concept, that every forecaster position is equipped with a SUN workstation for graphical applications and a PC for text applications.

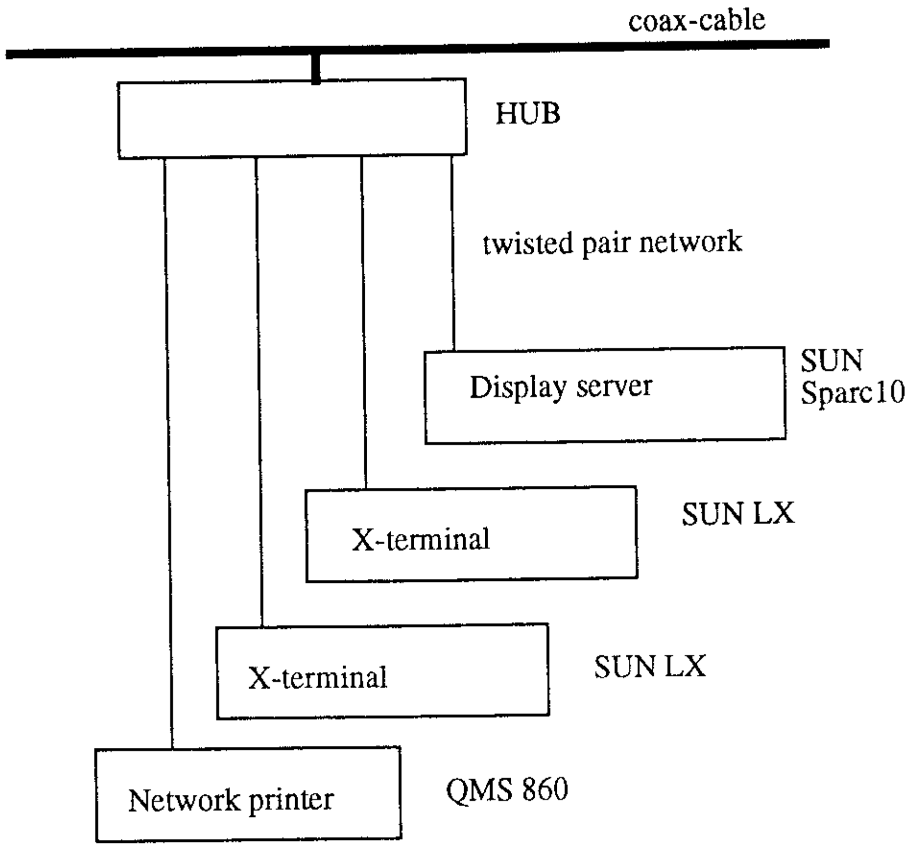
To reduce system administration and get a better performance we decided to set up the workstations as X-terminals. Tests showed us that X-terminals often have a very limited memory and that they can crash when all memory is used. We found no X-terminals that were able to swap to a disk like a workstation does, but a workstation with only a few background processes can perform just as well as an X-terminal.

We have 6 SUN LX with 64 MB memory running as X-terminals connected to 3 SUN Sparc10 display servers.

In an attempt to reduce the load on the network caused by X-terminals, the network is segmented with a display server and its X-terminals on the same segment. The idea being that network traffic between X-terminal and display server should not be routed to other segments.

We also have som standalone workstations, to make sure that not everything is depending on the network. All workstations can run all the applications, but since the different forecaster positions have different needs only part of the applications are selected for each workstation.

Segmented network



Solaris 2

When we started installing new hardware SUN shipped Solaris 2.1 as the only possible operating system for the LX, so we were forced into the change earlyer than expected. Solaris 2.1 was not a succes , to be polite...

Solaris 2.2 came soon after and it prooved to be a lot better, but still not perfect and a lot of patches had to be installed.

Porting our applications to Solaris 2 was not the problem. Most of them could run in binary compatibility mode or just needed a recompilation.

The big problem with Solaris 2 is that system administration has changed completely. Setting up users, printers or file systems is totally different. The print system has been changed, and since we use the print spooler system for distributing data that has given us a lot of problems and a lot of patches from SUN.

At the moment we use Solaris 2.3 and that has solved most problems, but patches are still plentifull.

Motif

As announced last year we have decided to use Motif for all new applications, and gradually we will be porting old applications for Motif as well. The first couple of applications were made without a GUI bulder, but that is a very tedius way of working.

In the written report on GUI builders we have stated the considerations on which GUI builder to use and we decided to go for XDesigner.

At the moment only a few applications have been developed using XDesigner, but so far we have been very happy with it.

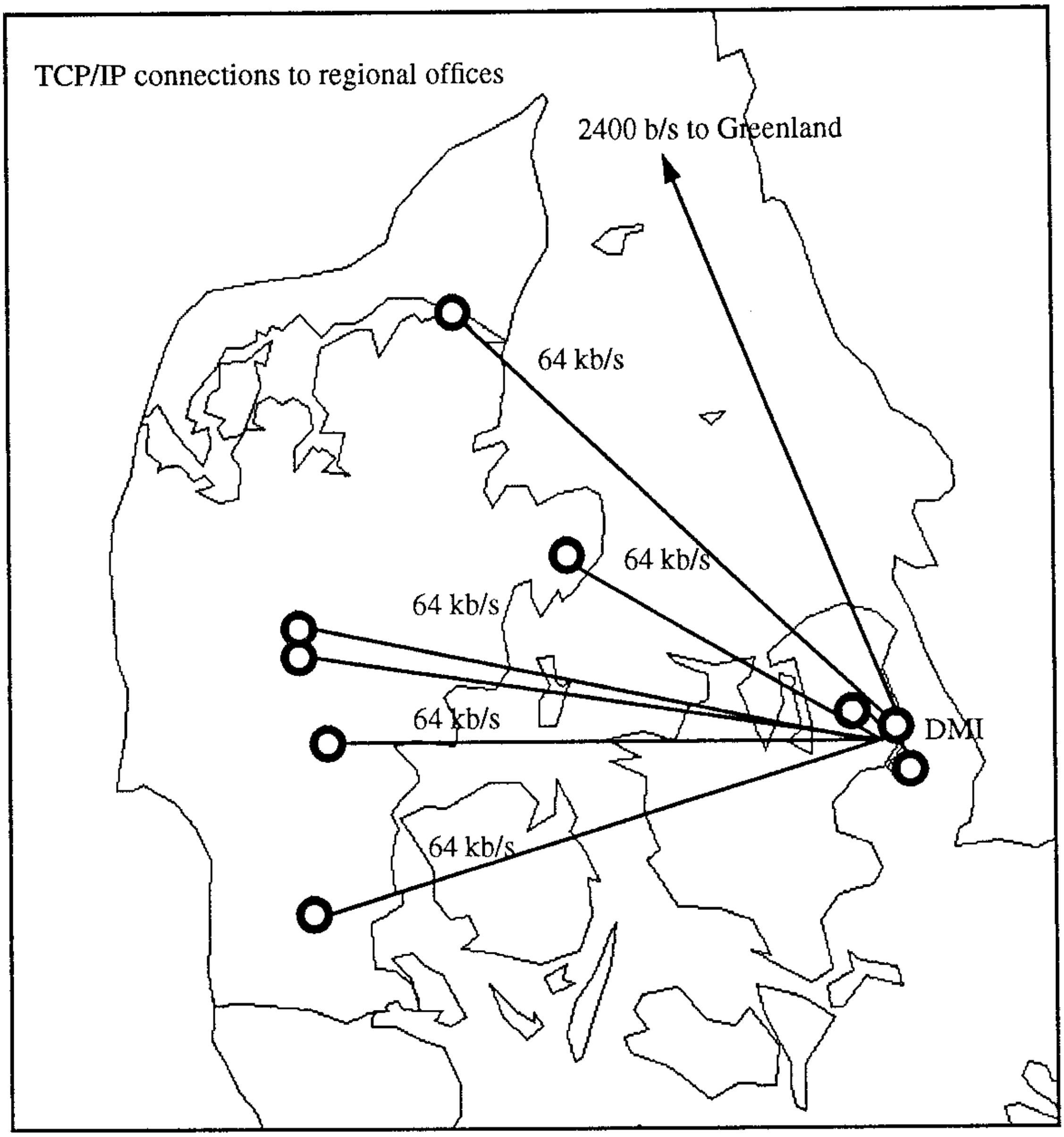
It has been discussed at DMI to make a styleguide for applications, but so far nothing has been put to paper. We try to follow the Motif styleguide, but not in all details.

Developments

There has been time for a few developments, the most innovative being the automatic system for generating forecasts for sea areas (described in a separate document).

SatPos	Shows the actual positions of NOAA satellites.
Farvand	User interface for automatic forecasts for sea areas. The forecasts use a predefined set of sentences and makes it possible to generate text in danish, english and german on an automatic phone system.
WindProfile	Presentation of data from the Wind Profile Radar.
NumModel	New version of our presentation of data from the numerical models with colours and animation. It shows precalculated PostScript images using Display PostScript introduced in Solaris 2.3.

Status of the regional offices



The workstations at the regional offices and military airfields have all been upgraded to Solaris 2.3. Communication runs on 64 kb/s lines from DMI directly to each regional office.

To Greenland the X.25 communication has been changed to a less expensive fixed 2400 b/s line. Running TCP/IP on a 2400 b/s line requires that only a few workstations are allowed acces to the line.

All regional offices are equipped with a SUN workstation with a PostScript laserprinter on the parallel port and a penplotter on a serial port. The laserprinter is for general use by applications and the penplotter is used for plotting of SYNOP observations.

Schweizerische Meteorologische Anstalt

Switzerland

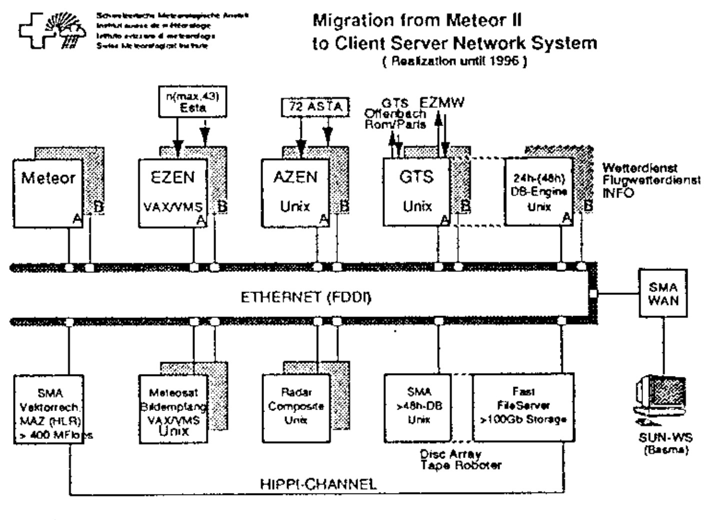
Swiss Meteorological Database using NEONS*

- Historical Overview
- System Requirements
- Solving the Problem
  - → Relational Approach
  - → Cyclic Approach
  - → Waste Table Approach
- NEONS
  - → Data Ingest
  - → Data Administration
  - → Data Retrieval
- Data Distribution

*Naval Environmental Operational Nowcasting System

Historical Overview

Jun. 91 Conceptual Design
Client Server Network System at SMI



Jun. 92 Initial Approach to Build a Database for Meterological Applications on RDBMS

Feb. 93 Contacts with NRL for NEONS

Sep. 93 Installation and Tests of NEONS 3.6

The Problem

How to build an effective Meteorological Information System having

- Binary, Text and Integer Data
- of a large Volume
- Arriving in a Continuous Stream

so that

- Information is Available Upon Arrival
- the System is Self-Monitoring
- Data is separated from Applications

Meteorological Data at SMI

Data	Max. Ingest Rate	Volume / Day
Messages (Decoded)	every 10 min.	~ 40 000 recs.
Messages (Coded)	continuously	~ 32 000 recs.
GRIB Data	twice daily	~ 10 000 fields
Volume Data (Model, Radar)	every 5 min.	1 728 images 14 200 fields
Radar Images	every 5 min.	288 images
Satellite Images	every 30 min.	144 images

Database Requirements

- Performance
  - → High Ingest Rate
  - → Fast Access
- Base for Meteorological Applications
  - → Superposition of Datasets
- Maintainability
  - → Constant Volume
  - → 24 hours - all year
- Reliability
- Ease of Extension and Modification
- Compatibility

Logical Database Design

- Parameter Data

Date + Time (TID), Site ID (SID), Parameter (P1, P2), Quality (Q1, Q2, Qn)

RP	TID	SID	P1	Q1	P2	Q2	Qn

1 Table for each Message Type

- Text Data

RM	TID	CCCC	TT	Text
				- METAR / SPECI - TAF - SIGMET - GAFOR

- Descriptive Data

RD	SID	Name	Latitude	Longitude	

Physical Database Design

- Retrieval Structure

  Index Sequential
  on Composed Key
  Time and Site Identifier

- Locking None / Record Level

- Shared Memory

  Global Buffers, Mapped File

- Data Administration

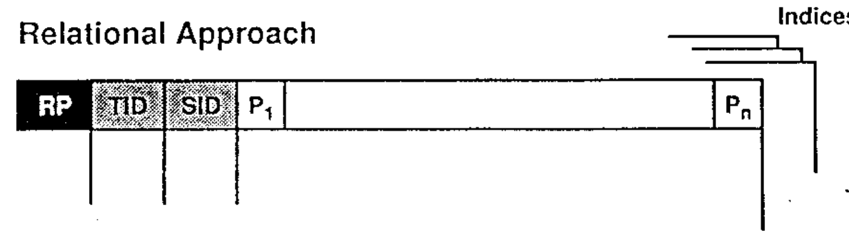
  Constant Number of Records:
  Delete Record (- 48 h)
  Insert Latest Record

- Index Administration

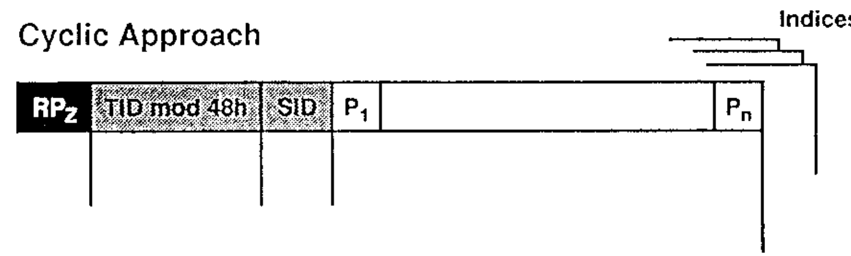
  Periodically Table Reorganization
  Alternative: Indices only on cyclic Key Attributes

Solving the Problem

Relational Approach



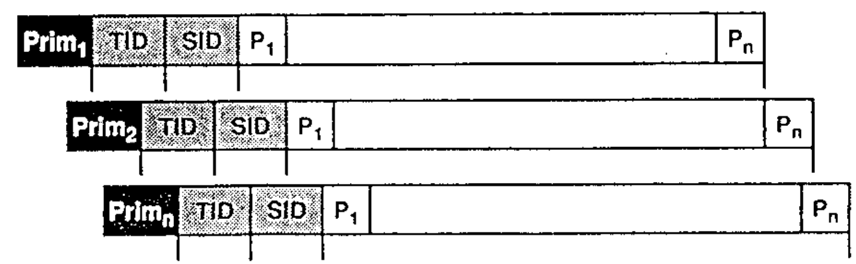
Cyclic Approach



24 h -Operation:

	RP	RP_Z	$Prim_1$
• Data Reorganization	-	+	+
• Locking	-	+-	+

Waste Table Approach



NEONS *

Set of Generic Geophysical Database Management System Software which Employs the Commercial Relational DBMS Technology to

- Perform Routine Database Housekeeping Functions
- Provide Instant Data Inventory List
- Standardize Data Access Interface
- Provide Data Preprocessing Utilities
- Store/Retrieve Data Sets across Network

Description by: Scott Christensen,
Naval Research Laboratory
Marine Meteorology Division
Monterey, CA, USA.

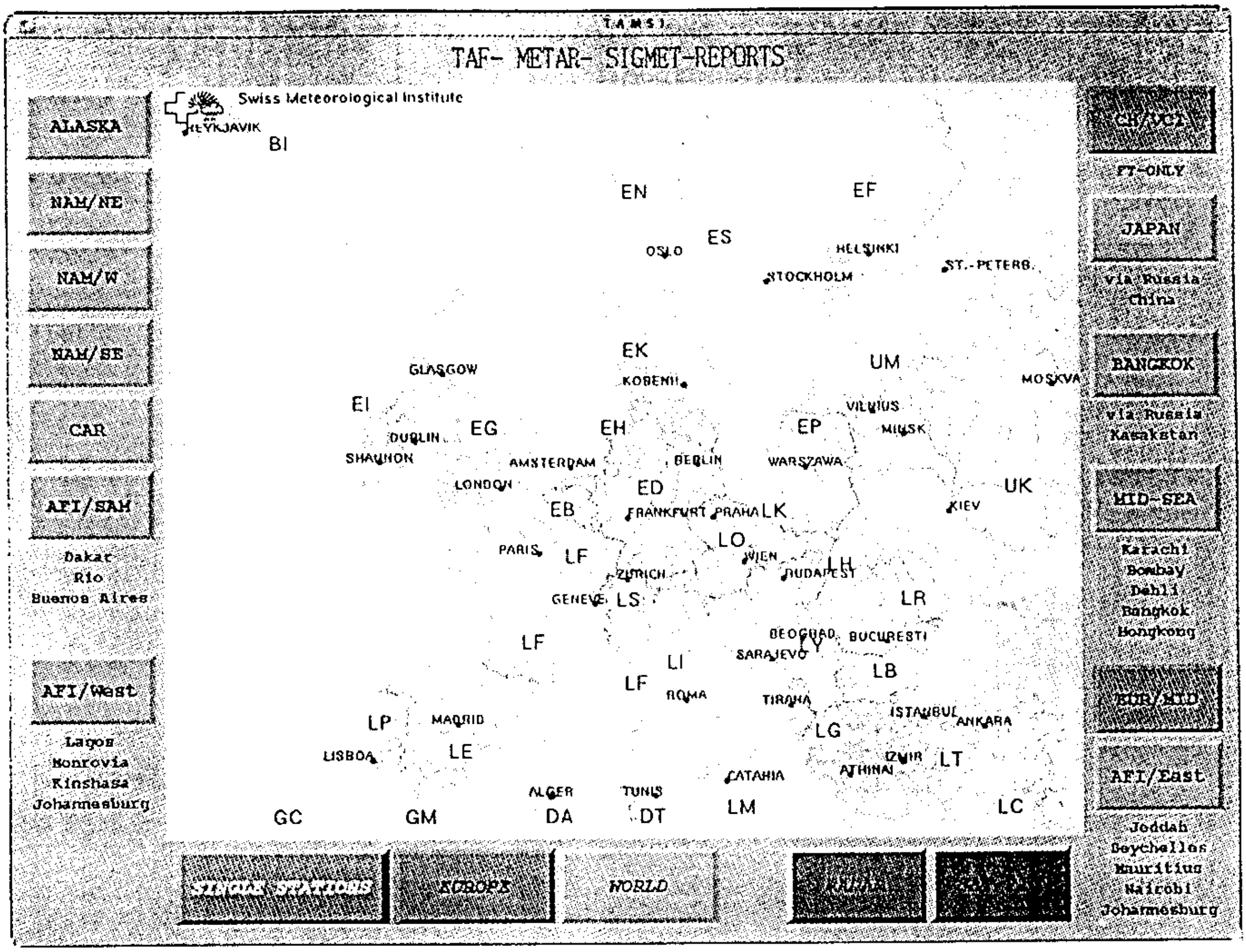
*Naval Environmental Operational Nowcasting System

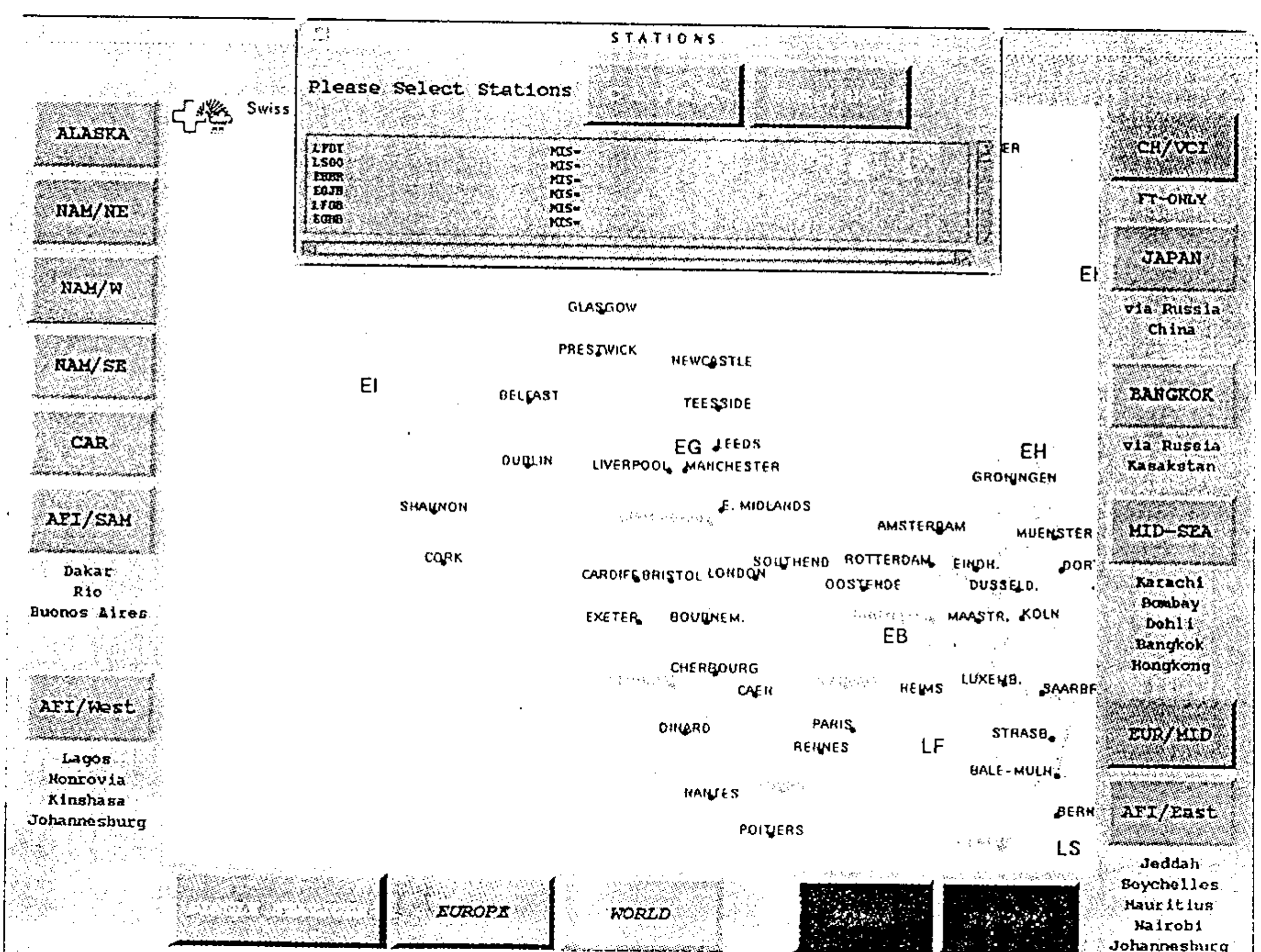
Generic Datatypes

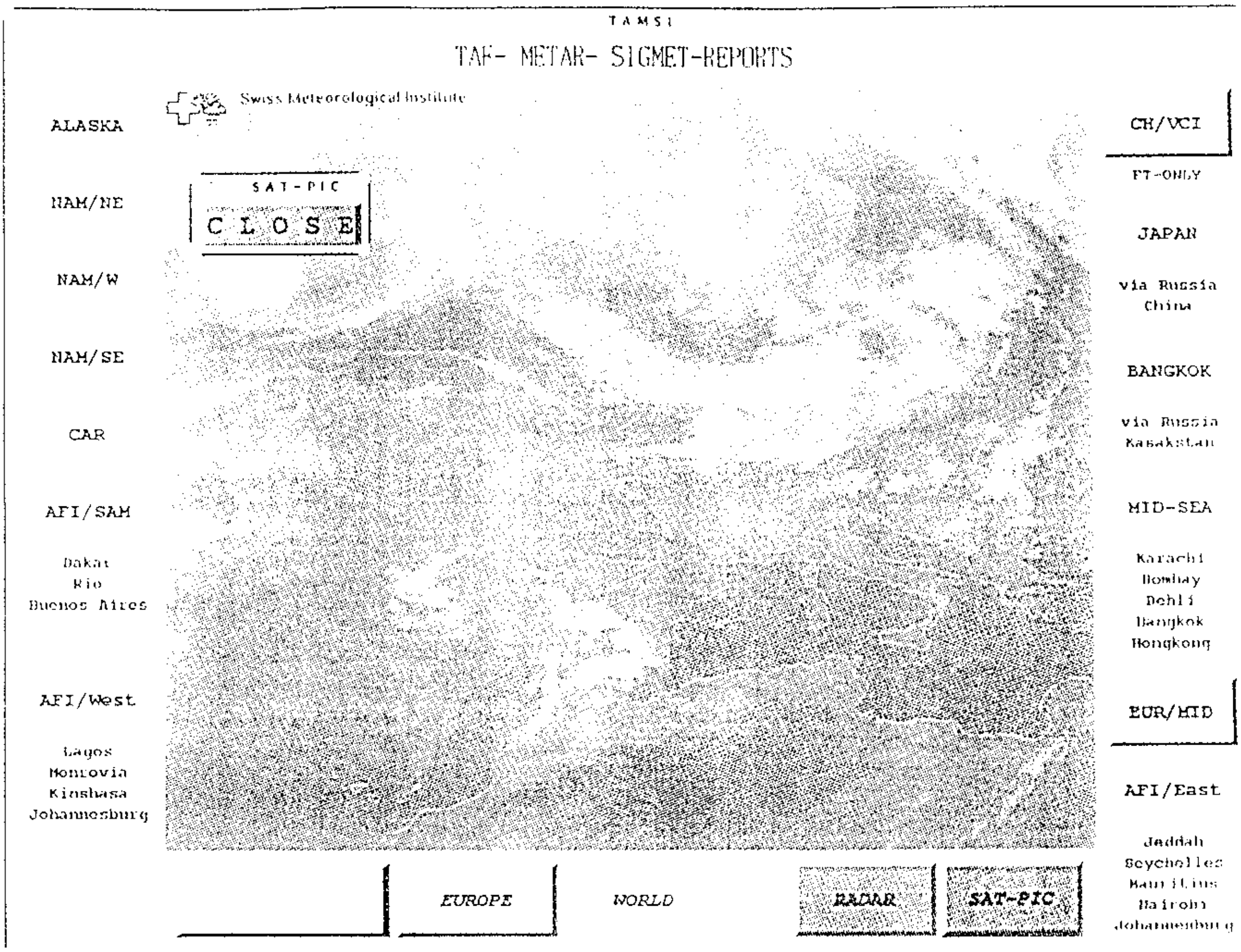
- Point: Latitude - Longitude - Time
- Line: Collection of Points
- Grid: Fields (Projections)
- Volume: Layers of Fields
- Image: Satellite and Register Coordinates

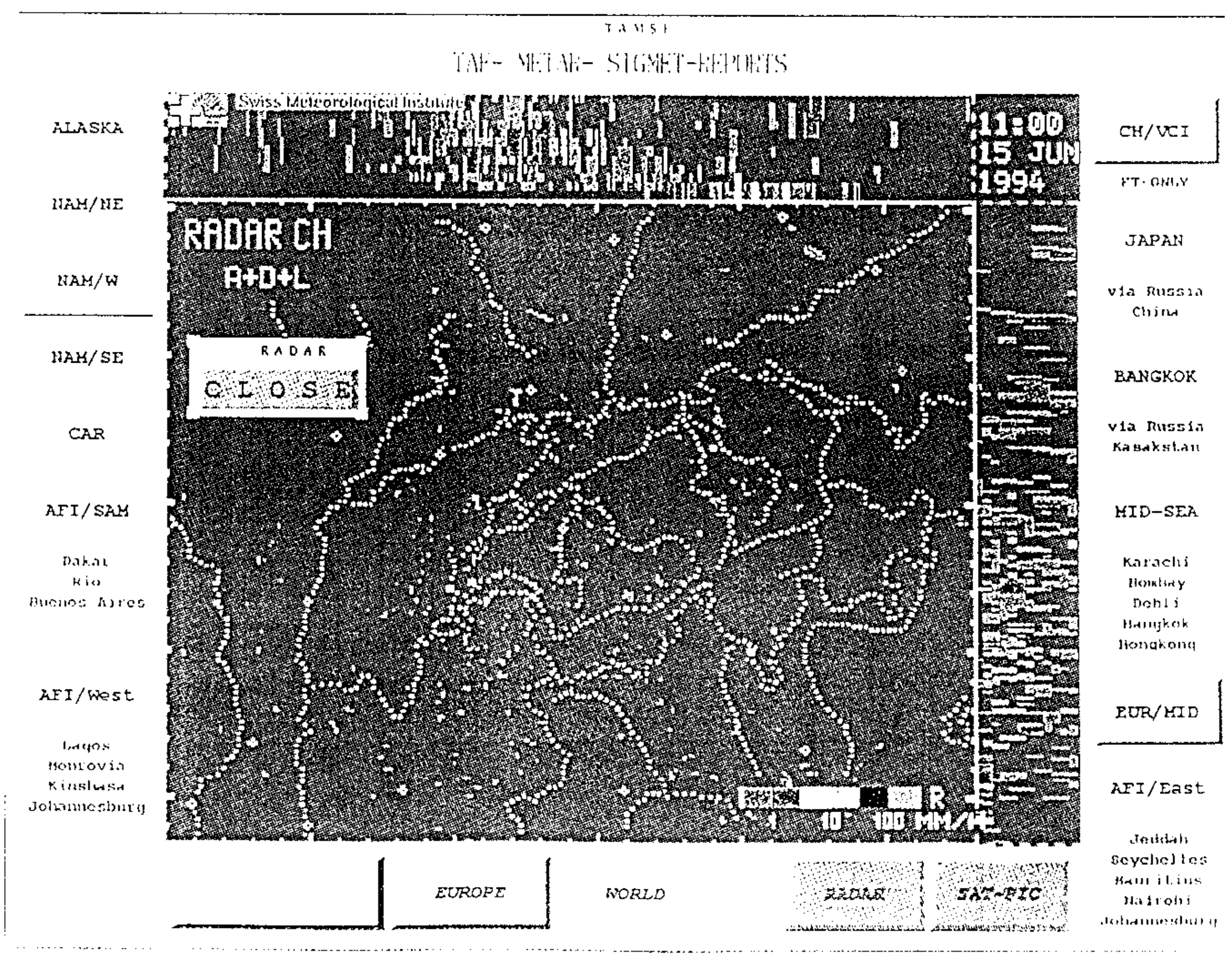
↓

- Simplifies Data Manipulation
- Simplifies Data Query
- Simplifies Extension and Modification
- Compact Program Libraries

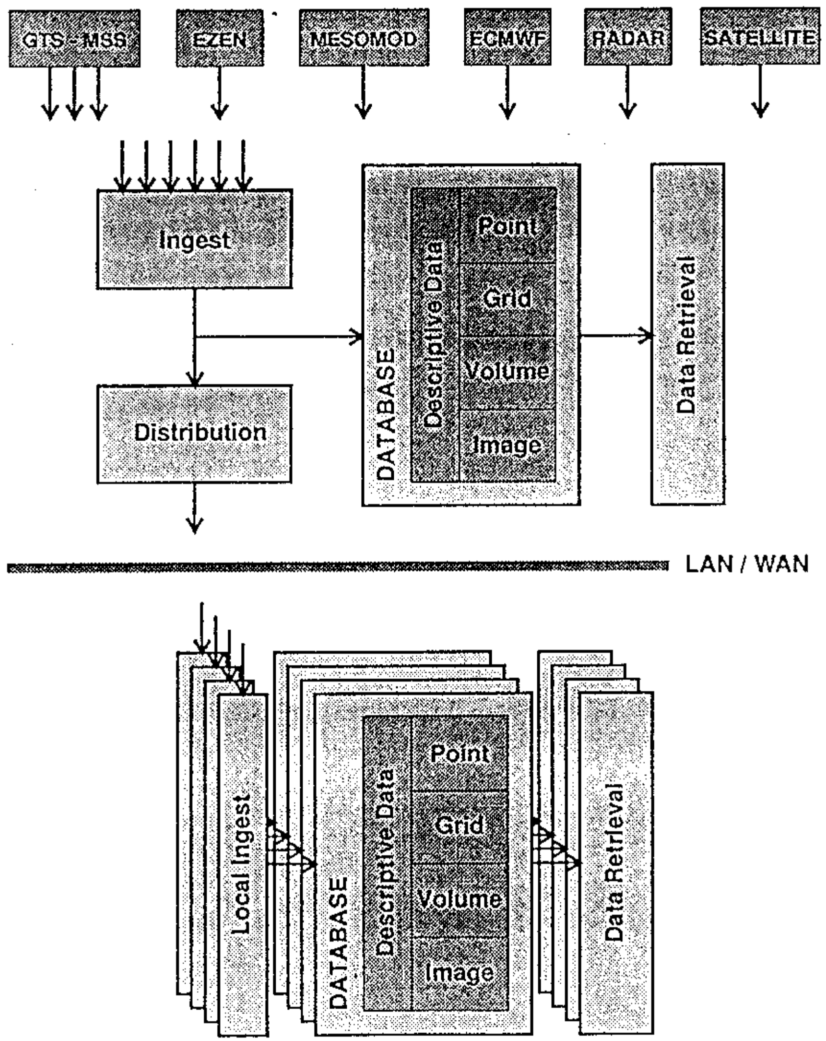

TAMSI
TAF- METAR- SIGMET-REPORTS
Swiss Meteorological Institute
ALASKA
NAM/NE
NAM/W
NAM/SE
CAR
AFI/SAM
Dakar
Rio
Buenos Aires
AFI/West
Lagos
Monrovia
Kinshasa
Johannesburg
CH/VCI
FT-ONLY
JAPAN
via Russia
China
BANGKOK
via Russia
Kasakstan
MID-SEA
Karachi
Bombay
Dehli
Bangkok
Hongkong
EUR/MID
AFI/East
Jeddah
Seychelles
Mauritius
Nairobi
Johannesburg
SINGLE STATIONS
EUROPE
WORLD
REYKJAVIK
BI
EN
EF
ES
OSLO
HELSINKI
STOCKHOLM
ST.-PETERB.
EK
UM
GLASGOW
KOBENH.
MOSKVA
EI
VILNIUS
DUBLIN
EG
EH
EP
MINSK
SHANNON
AMSTERDAM
BERLIN
WARSZAWA
LONDON
ED
UK
EB
FRANKFURT
PRAHA
LK
KIEV
PARIS
LF
LO
WIEN
LH
ZURICH
BUDAPEST
GENEVE
LS
LR
LF
BEOGRAD
BUCURESTI
SARAJEVO
LY
LI
LB
LF
ROMA
TIRANA
LP
MADRID
ISTANBUL
ANKARA
LG
LISBOA
LE
IZMIR
LT
ATHINAI
ALGER
TUNIS
CATANIA
GC
GM
DA
DT
LM
LC


STATIONS
Please Select Stations
LFBI MIS=
LSGG MIS=
EBBR MIS=
EGJB MIS=
LFGB MIS=
EGBB MIS=
Swiss
ALASKA
NAM/NE
NAM/W
NAM/SE
CAR
AFI/SAM
Dakar
Rio
Buenos Aires
AFI/West
Lagos
Monrovia
Kinshasa
Johannesburg
CH/VCI
FT-ONLY
JAPAN
via Russia
China
BANGKOK
via Russia
Kasakstan
MID-SEA
Karachi
Bombay
Dehli
Bangkok
Hongkong
EUR/MID
AFI/East
Jeddah
Seychelles
Mauritius
Nairobi
Johannesburg
EUROPE
WORLD
GLASGOW
PRESTWICK
NEWCASTLE
EI
BELFAST
TEESSIDE
EG
LEEDS
EH
DUBLIN
LIVERPOOL
MANCHESTER
GRONINGEN
SHANNON
E. MIDLANDS
AMSTERDAM
MUENSTER
CORK
SOUTHEND
ROTTERDAM
EINDH.
CARDIFF
BRISTOL
LONDON
OOSTENDE
DUSSELD.
EXETER
BOURNEM.
MAASTR.
KOLN
EB
CHERBOURG
CAEN
REIMS
LUXEMB.
DINARD
PARIS
RENNES
LF
STRASB.
BALE-MULH.
NANTES
BERN
POITIERS
LS


TAMSI
TAF- METAR- SIGMET-REPORTS
Swiss Meteorological Institute
ALASKA
NAM/NE
NAM/W
NAM/SE
CAR
AFI/SAM
Dakar
Rio
Buenos Aires
AFI/West
Lagos
Monrovia
Kinshasa
Johannesburg
SAT-PIC
CLOSE
CH/VCI
FT-ONLY
JAPAN
via Russia
China
BANGKOK
via Russia
Kasakstan
MID-SEA
Karachi
Bombay
Dehli
Bangkok
Hongkong
EUR/MID
AFI/East
Jeddah
Seychelles
Mauritius
Nairobi
Johannesburg
EUROPE
WORLD
RADAR
SAT-PIC


TAMSI
TAF- METAR- SIGMET-REPORTS
Swiss Meteorological Institute
11:00
15 JUN
1994
RADAR CH
A+D+L
RADAR
CLOSE
ALASKA
NAM/NE
NAM/W
NAM/SE
CAR
AFI/SAM
Dakar
Rio
Buenos Aires
AFI/West
Lagos
Monrovia
Kinshasa
Johannesburg
CH/VCI
FT-ONLY
JAPAN
via Russia
China
BANGKOK
via Russia
Kasakstan
MID-SEA
Karachi
Bombay
Dehli
Bangkok
Hongkong
EUR/MID
AFI/East
Jeddah
Seychelles
Mauritius
Nairobi
Johannesburg
EUROPE
WORLD
RADAR
SAT-PIC

Data Distribution



Conclusion

Using NEONS we can:

Provide a Database to Support Meteorological Applications with Real-Time Information around the clock

Next Step

Designing and Building a General Graphical User-Interface to Visualize the Information

↓

Meteorological Workstation at SMI

An tSeirbhís Mheitéareolaíochta

Ireland

Irish Meteorological Service: Report to EGOWS-5 [1994] Meeting

(J.Hamilton, Meteorological Service, Glasnevin Hill, Dublin 9, Ireland)

Introduction

I attended the EGOWS meeting last year and gave a fairly detailed account of the software and hardware configuration at the Irish Meteorological Service [Hamilton, 1993]. Consequently, this year, I will just summarise this data and I will concentrate on new developments and plans for the future.

Overview of the Hardware

We have quite a mix of hardware but at present we are moving more and more towards unix platforms. We have a number of dedicated servers and the current configuration consists of [see Figure 1]:

--A VAX cluster consisting of a MicroVax 3100 and two VAX 4200 used for telecommunications

--A MIPS RS-4000 Millenium used for decoding observations and NWP products and for the graphical display of such products. [An older, and less powerful, MIPS RS-3000 Magnum operates in 'stand-by' mode and is used as a backup in the case of a breakdown].

--A DEC-System 5000/200 server and 10 DEC-station 5000/120 workstations running INGRES [a commercial data-base package] used by the Climatological Division.

--Two networked Apple Macintosh computers used for desk-top publishing.

--Various networked PC's [running MS-Windows] used for word-processing etc.

--A VAX-server 3300 and three VAX-station 3100 workstations used for displaying the output of the Ericsson doppler radar at Dublin Airport.

--A Silicon Graphics Challenge-L server [with two 150Mhz processors] which is used for running the HirLam NWP model. The HirLam model has been running daily since January and it is planned to make the system operational within the next three months.

--A Silicon Graphics Indigo R3000 workstation and 4 Silicon Graphics Indy R4000 workstations used by the Research and Computer Divisions. Their primary use is to display output from numerical models. The Indy workstations were purchased late last year.

Meteorological Service, Glasnevin Hill, Dublin 9

LAN and Computer Configuration
(10Base-5,10Base-2 and 10BaseT Ethernet)

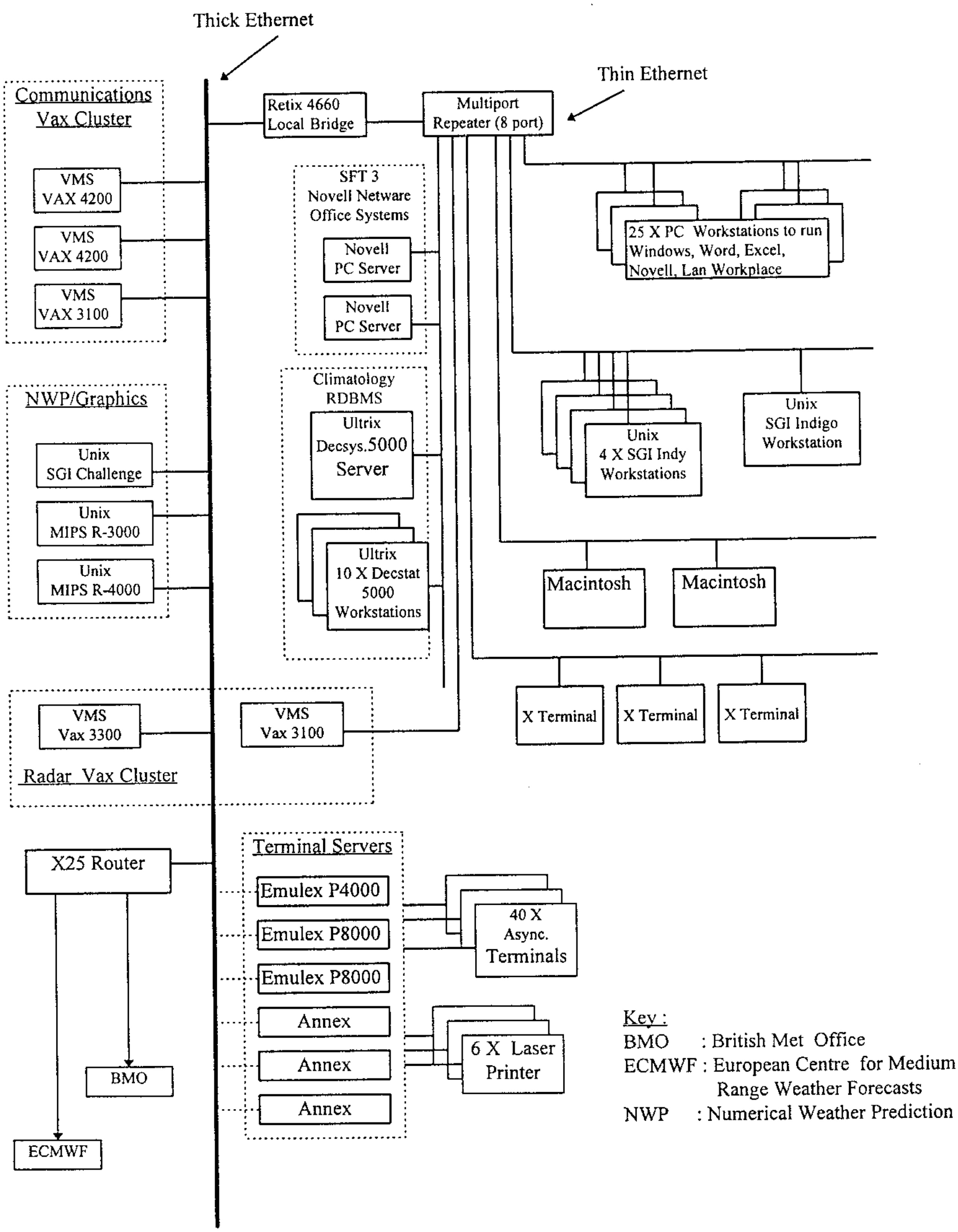


Figure 1: The computer configuration at the Irish Meteorological Service [June-1994].

It is planned to purchase additional workstations for the forecast offices in the near future. These will be used primarily to display output of numerical models [including ECMWF and HirLam].

Overview of the Software

There are a number of computer graphics packages in use in the service -- some are commercial systems and some were developed in-house. Generally speaking, the various packages are not well integrated and it is hoped to have a much more integrated system in the future. The following is a summary of the various packages [for further details see Hamilton, 1993]:

--We have two digitised radar stations in Ireland located at Dublin and Shannon Airports. Both radars are linked into the COST radar network and various composite plots are available on PC displays in the forecast offices. The Dublin Airport radar is quite new and was supplied by Ericsson. The system includes VAX-station workstations in the forecast offices with numerous display options including time loops, Doppler winds and vertical cross-sections. The Shannon Airport radar is quite old and it is planned to replace it this year with a Doppler radar and a new display system. Ideally we would like a software solution which will integrate the Dublin and Shannon radars.

--There are a number of self-contained satellite display systems which use secondary data. We plan to purchase a PDUS system, to receive primary data, within the next 2-years. However, we have made no definite decision on the type of system we will use, since we are concentrating on the purchase of the new Shannon radar.

--The in-house developed batch graphics systems include a package to display observations and/or NWP products in hardcopy form on pre-printed sheets using on-line Hewlett-Packard Draftmaster plotters. In addition, plots can be produced using Canon laser printers. We have just purchased a number of Hewlett-Packard PostScript printers and Figures 2 and 3 compare the output produced with the two types of printer.

--Plots destined for the Hewlett-Packard plotters [or the laser printers] are stored as random access binary files containing vectors. Each file may contain one or more plots [e.g. a file could contain a number of plots of geopotential at various standard levels]. Packages exist to display such plot-files on any of the available output devices which include the Hewlett-Packard plotters, the Canon laser printers, the Hewlett-Packard PostScript printers, the DEC-REGIS terminals [VT-340's] and various X-windows workstations.

--The forecasters also have access to an on-line interactive graphics system [called CHARTS] which allows the display of NWP products on a colour graphics terminal [a DEC VT-340]. CHARTS uses a command language which has been designed to be as easy to use as possible. Thus the forecaster can request plots with commands like the following:

```
PLOT 24HOUR ECMWF SURFACE PRESSURE
PLOT 2DAY 500MB GEOPOTENTIAL
PLOT 24HOUR 1000-500MB GEOPOTENTIAL
```

Full details of the CHARTS command language are given in Hamilton

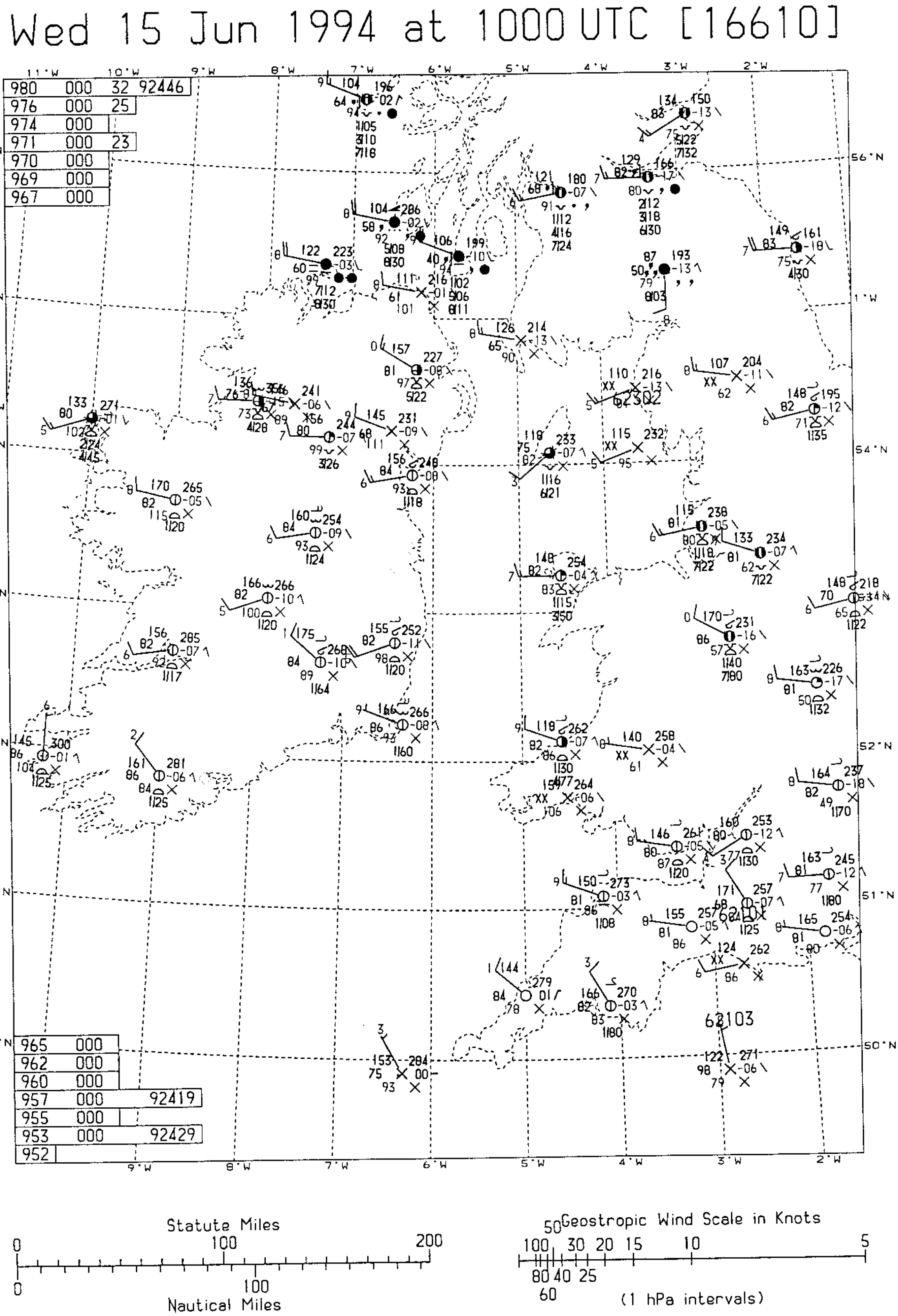


Figure 2: A plot of hourly observations produced on a Canon laser printer. This plot illustrates the use of the SYNOP plotting routines and the background map routines [note the latitude longitude lines are clipped at the coastlines].

Fri 17 Jun 1994 at 0800 UTC [16808]

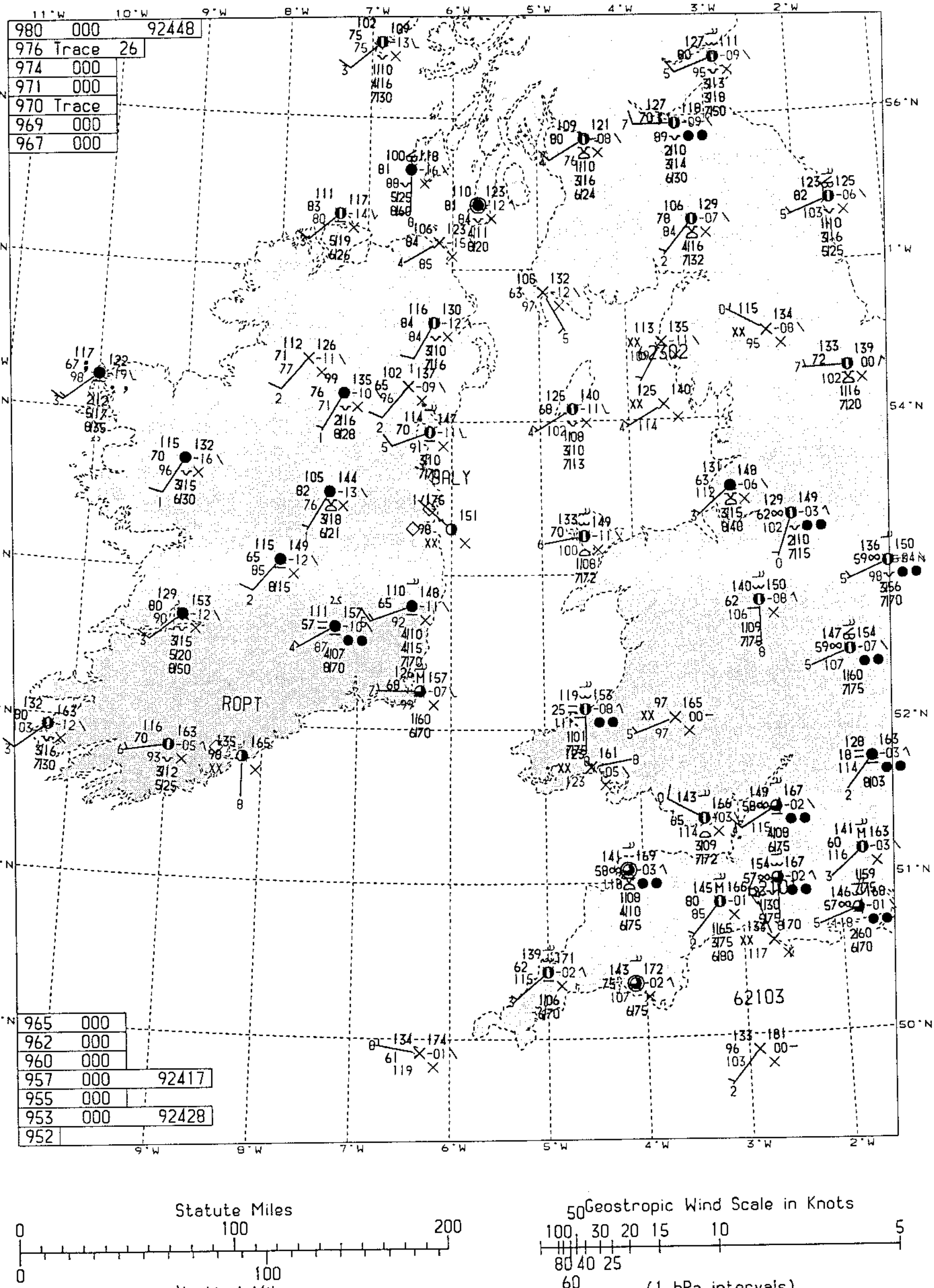


Figure 3: A plot of hourly observations produced on a Hewlett Packard PostScript laser printer. This plot is similar to Figure 2. Note the use of hardware polygon shading for the background map.

[1984]. Note that the third example plots a thickness chart.

--The climatological section produces a monthly climatological newsletter using an Apple Macintosh and the Quark Xpress desk-top publishing system. Many of the graphs and contour plots in the bulletin are produced using our own graphics packages. An in-house developed system is used to analyse and draw shaded climatological maps of mean monthly temperature, mean monthly sunshine and total monthly rainfall.

--We have implemented the ECMWF MAGICS package and it is used regularly for special projects. We have two implementations: the first, on a MicroVax, uses own [minimal] version of GKS; the second, on an SGI workstation, uses Xelion GKS.

--We are looking at the possibilities of implementing the Hirlam MetGraf package on a workstation. This package displays HirLam output and, in particular, it supports the generation and plotting of many derived products.

The Irish graphics system [including a contouring package, a package for displaying background maps, various packages for plotting observations and the CHARTS program] was originally developed on a DEC-2050. The various components have been ported to the VAX's and the various unix workstations/servers. All the software [with the exception of the X interface] is written in standard Fortran-77. The system has its own drivers for HP-GL [Hewlett Packard Graphics Language] and for PostScript.

Use of X-windows / Motif

Last year we described the development of a package, written in C and Fortran-77 using Xlib, to display plot-files on an X-terminal. Since then we have stopped development using 'raw' Xlib and we are now using Motif. Currently we have written three applications:

--plotxw: This is a package which can display a plot-file [produced using the 'batch' plotting system] on an X-terminal. The package includes all the functionality of the older package based on Xlib [including the zoom option] and, in addition, includes a hardcopy option.

--xgrbplt: This program is used for plotting NWP GRIB-code output from the HirLam model. The user can superimpose observations on the plot. This package is mainly used by the Research Division.

--xcharts: This package is still under development. It is planned as a replacement for the CHARTS program. It is hoped to retain as much compatibility as possible between the old and the new systems.

All these systems are written in a combination of C and Fortran-77. The main routine, which handles the Motif widgets and the various call-backs, is written in c; this then calls various Fortran packages [such as the contouring package]; and finally the Fortran packages call low level C routines [such as XDrawLines and XFillPolygon] to produce the actual output. The C code was written directly rather than using a 4GL tool [such as X-Designer].

We discuss these various packages in the following sections.

plotxw: Display plot-files using X-windows

As mentioned previously, plots destined for the Hewlett-Packard plotters [or the various laser printers] are stored as random access binary files containing vectors. Each file may contain one or more plots. plotxw can display such plot-files. The user gives the command:

```
plotxw plotfile.plt
```

where 'plot-file.plt' is the name of such a file. If the file contains just one plot then the plot is displayed and the user is presented with a menu, along the top of the chart, with the following options:

```
File, Bgnd, Zoom, Plot, Hard, Help
```

The 'File' button calls up a menu which includes the 'quit' option; The 'Bgnd' button allows the user to display the plot on a map corresponding to one of the pre-printed backgrounds used with the Hewlett-Packard plotters; the 'Zoom' button allows the user to select a rectangular zoom area by defining its two opposite corners using the mouse; the 'Plot' button cancels the zoom and displays the whole chart; the 'Hard' button produces a hardcopy [of the area on the screen which may be a zoom area]; and finally the 'Help' button produces some help text. The user can cascade zooms but the 'Plot' option will always return to the entire plot [not the previous zoom level]. When specifying a zoom the cursor appears as cross-hairs. The system preserves the aspect ratio of the plot during the zoom.

If the plot-file contains a number of plots then the menu contains additional entries and looks like this:

```
File, List, Bgnd, Zoom, Next, Prev, Plot, Hard, Help
```

The 'List' button allows the user to list details of the plot-file entries [including the size of each plot] and then select a plot from the list; The 'Next' button moves to the next entry in the list; the 'Prev' button moves to the previous entry. If the user selects a zoom it remains in place as the user moves back and forth between plots within the plot-file. The program starts with a blank screen -- the user must select the first plot for display.

xgrbplt: Display GRIB-files

This package is used to display GRIB output files form the HirLam model. It is mainly used by the Research Division. A HirLam GRIB file typically contains all the output data for a given time level i.e. it usually has a number of surface fields [such as msl-pressure, rainfall etc.] and a number of multi-level fields [e.g. temperature, geopotential, wind components etc.]. The user runs the package by giving the command:

```
xgrbplt fc9406150024pp
```

where 'fc9406150024pp' is the HirLam 24-hour forecast [from 15-June-1994] after post-processing onto pressure levels [the system can also plot model level files].

The program starts by reading the GRIB file and determining the grid geometry. Then it draws a polar-stereographic map which just covers the grid. The user is then presented with a menu:

File, List, Optn, Zoom, Next, Prev, Plot, Hard, Help

The 'File' menu button includes the 'quit' option; the 'List' option allows the user to select a field for plotting; the 'Optn' button [discussed later] is used to select overlay options; the 'Zoom' button is used to specify a zoom using the mouse; the 'Next' and 'Prev' buttons move to the next and previous plot in the list, respectively; the 'Hard' button produces a hardcopy and the 'Help' button produces some help text.

The 'List' button produces a list of options such as the following:

```
   6 100  200 Geopotential on Isobaric surface
   6 100  300 Geopotential on Isobaric surface
   6 100  500 Geopotential on Isobaric surface
   ...
   1 103    0 Pressure on Specified altitude
3334 105   10 Wind on Specified height level
  11 105    2 Temperature on Specified height level
  61 105    0 Total precipitation on Specified height level
  62 105    0 Large scale precipitation on Specified height level
  63 105    0 Convective precipitation on Specified height level
  71 105    0 Total cloud cover on Specified height level
  66 105    0 Snow depth on Specified height level
```

Note that this list is produced automatically by reading the file. The three numbers at the start of each line are the WMO parameter code, the WMO height code and the height value. Hence, the first entry specifies the 200mb geopotential, the fourth entry is the msl pressure etc. The package combines the u- and v- wind components [i.e. parameters 33 and 34] to produce a plot of WMO wind arrows.

The package allows the user to resize the window. When the window is resized the plot is redrawn i.e. re-contoured and a new background map generated. This is to allow for changes in resolution due to the resizing: the labels on the plot remain at constant size in terms of pixels.

The zoom option uses a cursor which follows the latitude/longitude lines i.e. the cursor appears as a circle intersected by a straight line on a polar-stereographic map [see Figures 4, 5 and 6]. Basically the user specifies two points on the chart and the zoom area is defined by the range of latitude and longitude defined by the points. A plot area is then calculated which includes the area requested by the user; the plot is realligned with the central meridian of the plot vertical. The package draws a label at the top of the plot produced by the zoom.

The user can plot difference charts with xgrbplt. The command is similar but two GRIB-files must be specified. They must be on the same area but need not contain the same number of fields or have the fields in the same order. The 'List' button will just show the fields common to both files.

The final way of using xgrbplt is to plot observations. An observation file [in a special format used as input to the Norwegian objective analysis system which we are using as part of HirLam] is specified on the command line. The result is that observations are displayed on the plots of geopotential and msl-pressure. The package uses a 'de-clutterring' algorithm to select the observations for display. The number of observations displayed varies with the size of the plot on

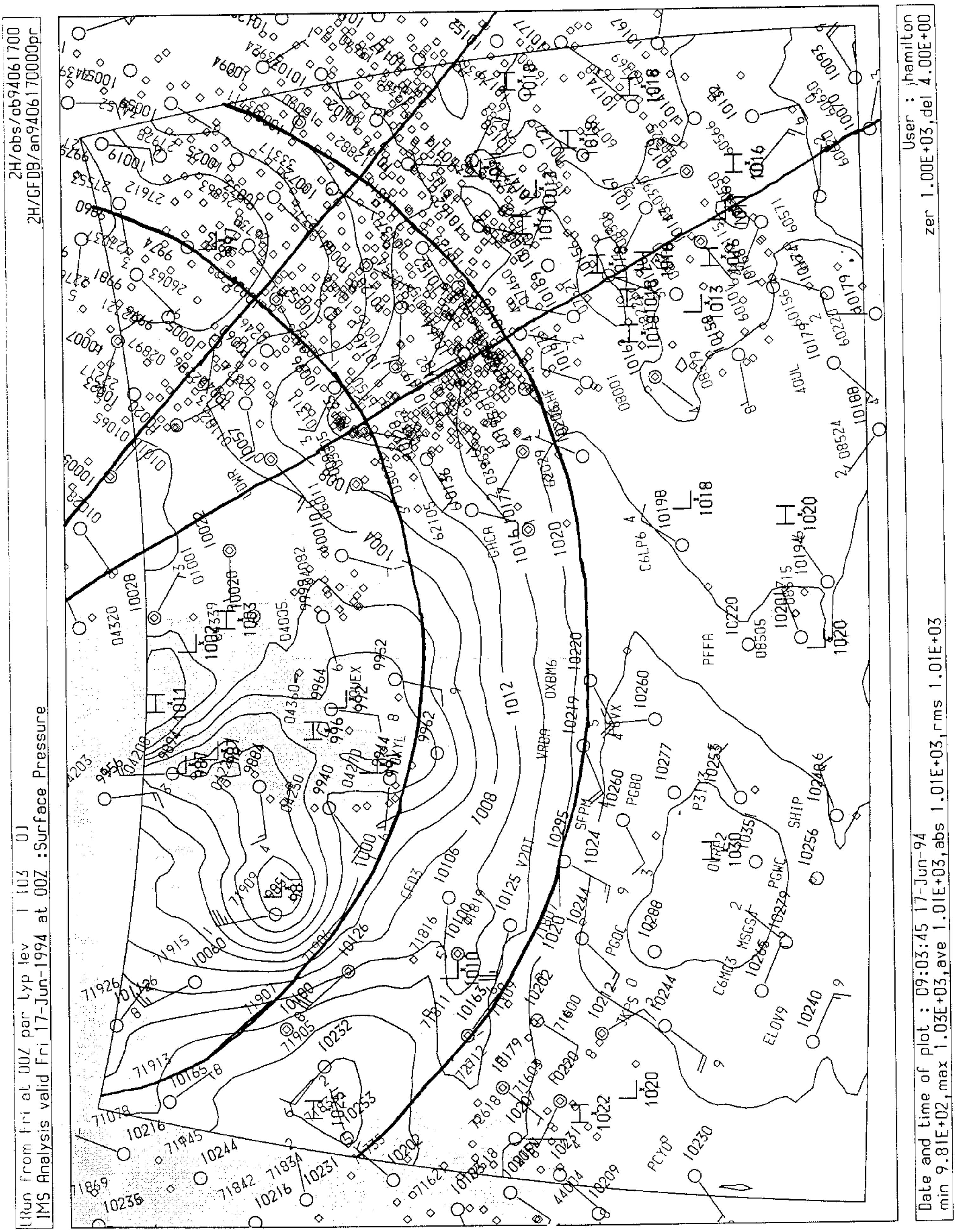


Figure 4: Hardcopy of a plot produced with the xgrbplt package. The lines drawn in by hand indicate the cursor used for zooming [see Figure 5]. Note that the cursor follows the latitude/longitude lines of the background map. The small diamonds indicate observations rejected by the de-clutterring algorithm.

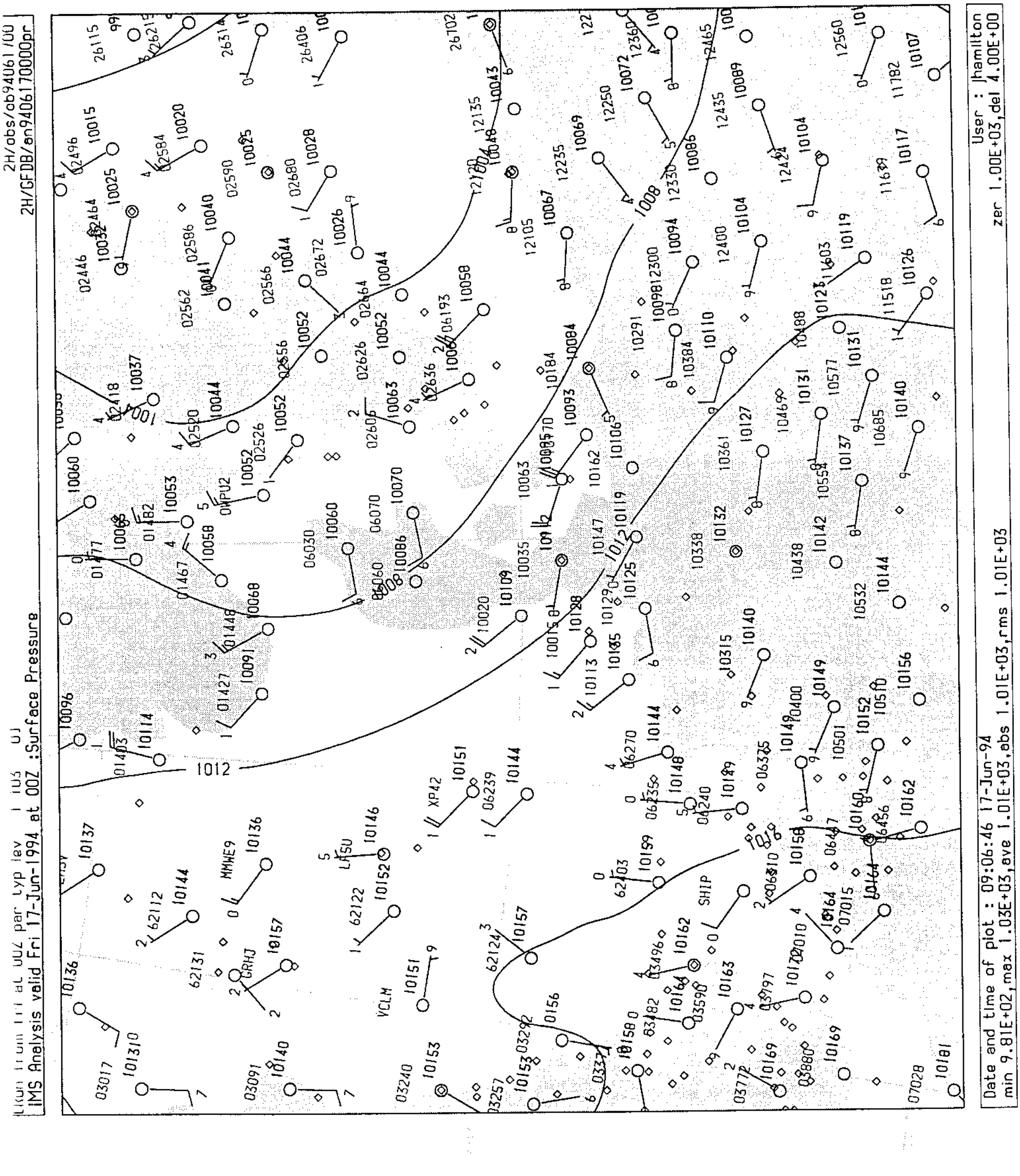


Figure 5: Hardcopy of a plot produced with the xgrbplt package. This chart is the result of the zoom applied to Figure 4. The map is choosen to include all the area specified by the zoom; the orientation of the chart is changed to give a vertical central medidian.

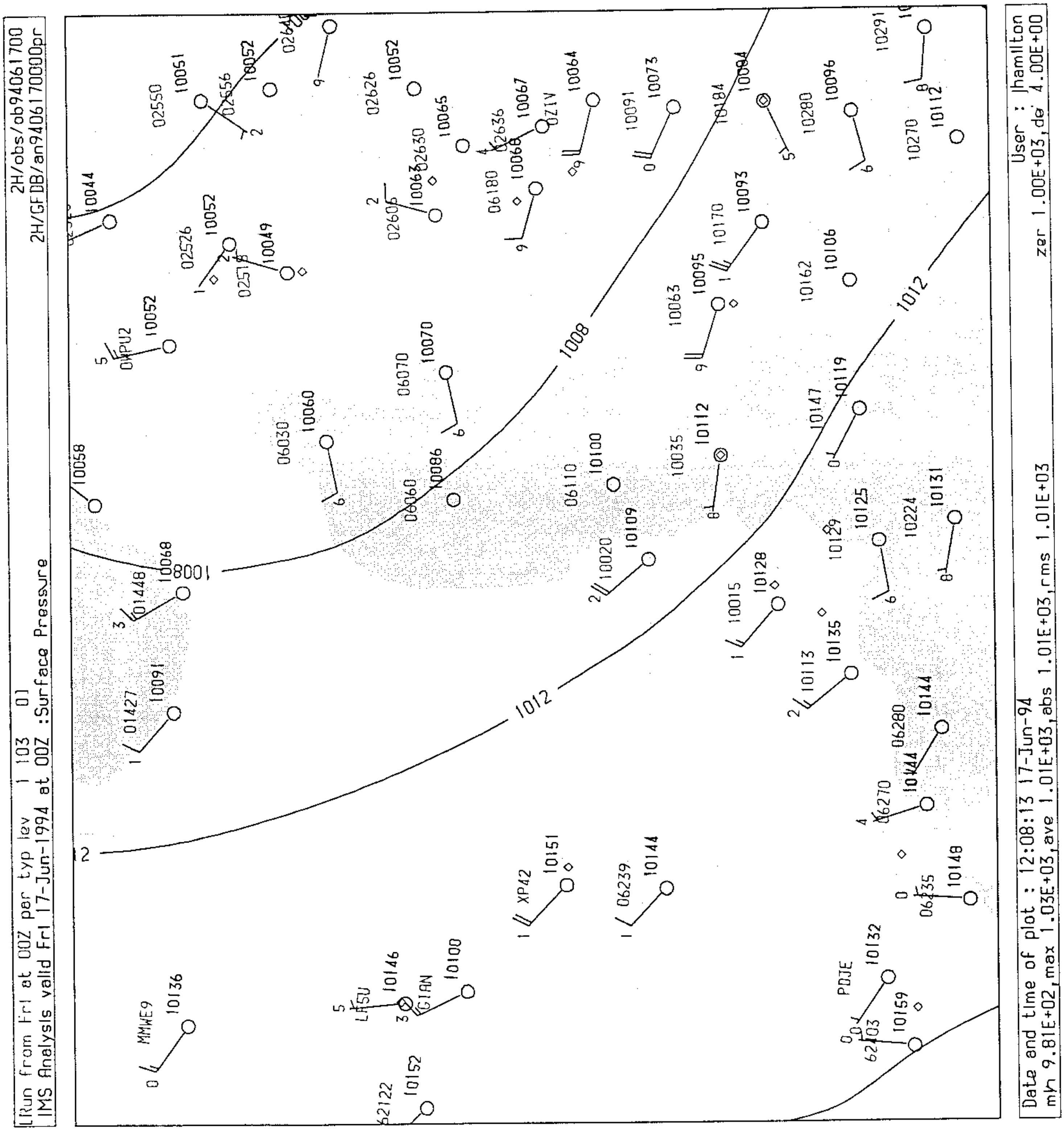


Figure 6: Hardcopy of a plot produced with the xgrbplt package. A further zoom has been applied to the chart in Figure 5. Note that, with a workstation, this plot is produced in a number of colours.

the screen and the zoom level. A problem with the present version of the package is that it does not preserve observations between zooms, i.e. each selection of observations is totally independent. It is planned to rectify this.

A typical plot can consist of a geographical background, a contour plot and a plot of observations. The 'Optn' button allows the user the following choices:

Field + Observations + Map
Field + Map
Observations + Map
Map Only

Figures 4,5 and 6 show hardcopies produced from this package. They illustrate many of the features discussed here.

xcharts: An X-windows version of CHARTS

CHARTS [Hamilton, 1984] is the main forecaster interface to NWP output. As mentioned above, it is a command driven interactive system which allows the display of charts on a DEC VT-340 terminal. It uses a command language which has been designed to be as easy to use as possible. Commands can be abbreviated. There is an on-line HELP system, a hardcopy option, a script option [viz. the so-called 'obey' files], and ambigious or incorrect commands produce meaningful error messages. The system remembers the parameters entered with previous commands and these become the defaults for subsequent commands -- this reduces typing to a minimum.

Using CHARTS the forecaster can access output from the [old] Irish Meteorological Service [IMS] model as well as the models of ECMWF and the UKMO. Available output includes plots of wave data. The forecaster can also request plots of upper-air ascents as tephigrams.

The forecasters are very familiar with the system and it is planned to build on this experience when designing the new system [to be called xcharts].

It is planned to have a user interface which combines a command line with menu buttons. This will allow for continuity between the old and new systems; it will also allow the use of the current set of script [i.e. 'obey' files]. Ideally, all features should be available with either the command interface or the menu interface.

The menu interface is likely to contain the following buttons:

... Parm, Fcst, Levl, Modl ... Plot, Over ...

which will allow the forecaster to specify, respectively, the parameter [e.g. geopotential], the length of the forecast [e.g. 24-hour], the level in the atmosphere [e.g. 500mb] and the model [e.g. HirLam]. After the specification is complete the user can then plot the product as a new plot ['Plot'] or overlay it on an old plot ['Over']. The various parameters can be selected in any order and parameters which have not changed since the last request need not be specified. Usually the user will select options with the left-hand mouse button and terminate the selection with the 'Plot' or 'Over' options. However, we are considering a number of short-cuts: one possibility might be for a 'double-click' to terminate a selection [and correspond to 'Plot'], another might be the use of another mouse button.

It should be clear from the above discussion that the user will be able to select non-existent products [e.g. HirLam 3-day forecasts are not available]. In such a case the system should print a warning message. It is difficult to prevent such impossible requests while making the system general and allowing the user to enter selections in any order. For example, suppose the current selection is 'ECMWF 10-day geopotential', and the user wants to select 'HirLam 36-hour geopotential'. The selection can be made in two ways: either select the forecast length and then the model or vice versa. In the first case the transition is:

'ECMWF 10-day' --> 'ECMWF 36-hour' --> 'HirLam 36-hour'

where all the intermediate options are legal. In the second case the transition is:

'ECMWF 10-day' --> 'HirLam 10-day' --> 'HirLam 36-hour'

where the request for the 10-day HirLam forecast is impossible. We are still considering this part of the design.

We are also considering methods of specifying difference charts. The present, command driven, CHARTS program allows arbitrary entries [e.g. the user can request a 1000mb-500mb thickness chart but it is also possible to request a 1234mb-456mb thickness chart - although the later is not available]. One solution is to present a standard set of thickness options and force the user to use the command line for less common ones; however this is not ideal and this aspect of the design is also being considered.

Finally, it is planned to incorporate a module into xcharts that will allow the plotting of observations. This will also necessitate the use of a zoom option. The zoom option is likely to use the 'latitude/longitude' cursor described in the section on xgrbplt.

Initially, not all the forecast offices will have workstations. Hence, the new version of xcharts should remain backward compatible with CHARTS to allow the use of VT-340 graphics terminals.

The short term plan is to have the initial release of xcharts operational by mid-September. This version will just display raw model output and some basic derived parameters. In the longer term it is hoped to include extra derived products [such as vorticity, potential temperature etc.] and we are looking at the HirLam MetGraf package in this context.

Conclusions

We have made some progress since the EGOWS meeting of last year: we how have SGI workstations and we have developed two applications [plotxw and xgrbplt] which are in use by the Research Division.

Workstations have not yet been installed in the forecast offices but work is progressing on a system [called xcharts] to display NWP output for the forecasters. It is hoped to purchase workstations for the forecast offices within the next few months and to have this system operational by about mid-September. This will coincide with the operational introduction of the HirLam model.

References

Hamilton, J.E.M., [1984]: 'The Design of an Interactive Graphics System for the Display of Meteorological Fields', Software Practice and Experience, Vol. 14, No. 6, p. 587-600 (June 1984).

Hamilton, J.E.M., [1993]: 'The Computer Graphics System used at the Irish Meteorological Service', European Group on Operational Meteorological Workstations [EGOWS], Deutscher Wetterdienst, Offenbach, June 1993.

Intitut Royal Météorologique de Belgique

Belgium

Meteorological Workstations at Belgian Royal Meteorological Institute

Luc Gerard, civ. Eng., IRM-KMI, Dept Applied Meteorology.

Software for operational meteorological workstations is intended for two classes of users:

-first, the operational forecaster, who is interested essentially in efficient routine products;

-second, the researcher who wants to develop application programs while benefitting of efficient and easy data access and graphical processing.

We can assert that the background processing needs of the operational forecaster embrace those of the developer, with addition that he wants to use a ready-made application layer.

In order to relieve the Meteorological Application Developer of the low level cooking tasks, we have to separe these from the applications. These low level tasks are:

-the data acquisition and database management,

-the meterological graphic drawing,

-the Graphical User Interface,

-the graphic post-processing tasks.

A graphic package like *MAGICS* was a first step on this way, taking care of the meteorological graphics and some data decoding.

Our own further development of MAGICS lead us to request the following features:

-integrate an interface to our data assimilation system into MAGICS, so that the application layer can directly ask the data requested through MAGICS functions;

-enhance and expand MAGICS' meteorological graphic facilities;

-integrate a complete GUI into MAGICS, allowing the application to interact with this GUI through simple MAGICS functions, or even without the application having to bother about it.

-develop graphic postprocessors, which can be used respectively for interactive editing, combination and plotting of *any graphical product generated through MAGICS*.

-keep the system sober in computing-power and disk space requirements.

With such facilities the researcher can concentrate his work on meteorology, while the compatibility is maintained between all applications and the portability is ensured between computers and between peripherals. Furthermore, the common tasks are done by shared routines, reducing computer load.

A. Data acquisition

-All data from the GTS are cleaned up, labelled and classified in ASCII files constituting the alphanumeric database, available to all applications;

-Certain of these data are then converted to BUFR format. We have choosen a BUFR message structure in which the *BUFR Message Subtype* in the identification section gives a direct and univocal information about the contents *and* the data structure. That means that for a certain subtype, the section 3 descriptors sequence is fixed. We are grouping several stations in one BUFR message, following their types: there is one subtype for the main synoptic stations, another for the secondary, a third for the mountain stations, etc.

-The GRIB data coming from ECMWF are also classified in a database, and some preprocessing is done to compute derived fields.

B. Enhanced Interactive Magics

1. Enhanced Structure

a. Data aquisition

Standard ECMWF BUFR and GRIB decoders are linked to the MAGICS library.

The data selection has been enhanced, in BUFR (see previous paragraph) and in GRIB (addition of the mode *IDENTIFY*, to locate the fields inside the GRIB file), we added the use of fields in polar-stereographic projection.

The application can still pass (and retrieve) FORTRAN arrays to MAGICS.

b. Drawing structuring

All drawings are structured in layers, each layer having a litteral name and beeing susceptible to be handled individually, as to be made visible or not, saved, deleted,...

c. Graphic data saving

The picture layers can be saved with their names, and the projection and page data into GKS metafiles. It is also possible to produce pixmap files in xwd format, easy to superpose to satellite pictures.

2. Interactivity

We used in our present developements a Motif-like interface *included in GKS*, and it was not necessary to develop more general XWINDOW possibilities; the big advantage in doing so is that the Application program has NOT to make calls to Motif NOR to complex local GUI routines: it *can actually leave all of that to MAGICS*, in which a very small number of Upper Level interactivity routines have been included. These are the following:

-Automatic interactivity routines:

CALL PZOOM: to produce a zooming phase with layers visibility selection;

CALL PEDIT: allows zoom, pan, and recording of positions of bad observations, that have to be rejected for a subsequent analysis;

CALL PSAVE: menu of layers selection and saving in a metafile.

-Application Programmable routines:

CALL PMENU ('1st choice, 2nd,   ,last.', ich): presents a menu and returns the interactive user's choice to the application.

*CALL PASKMI('Magics_parameter', *), CALL PASKMC( ), CALL PASKMR( )*: propose the current value of a MAGICS parameter to the interactive user, allowing him to modify it.

*CALL PASKI (Prompt-string, ivar, *), PASKR( ), PASKC( )*: propose the current value of a variable to the interactive user, allowing him to modify it.

CALL PMSG('Message'): gives a message to the interactive user.

Furthermore, *there is nothing to modify to the code* when changing from an OUPUT/INPUT category output device to an OUTPUT-ONLY device: the application programmed dialog will then automatically be produced on the TTY, while the pure interactive possibilities like ZOOMING or layers visibility menus will be skipped.

The use of specification files is also a very flexible way of tuning a big mass of parameters, while limiting the fully interactive possibilities to the strict necessary.

3. Graphic Capabilities

The main additions to the original MAGICS are the observations analysis in order to draw isolines, smoothing procedures, emagram drawing, and several smaller enhancements.

C. Interactive Graphic Post-processor

Any drawing produced by any application program using MAGICS can be viewed and edited by this post-processor. The main features are:

-constructed only on GKS and Fortran 77.

-Full interactivity: zooming, panning, geographical coordinates, layers names are available.

-Combination of drawings and manipulation of layers:

-selection, saving, deletion, renaming of layers;

-recolouring of layers while loading drawings;

-superposition/juxtaposition of layers on the graphic screen.

-Hand drawn items: Fronts, zones, lines, arrows, texts, meteorological symbols with different colours, sizes, line thicknesses and styles, fonts...

D. Plotting Post-Processor

-Can be called from the Interactive Graphic Postprocessor, as well as from the Operational Application Layer, or on the *unix* command line.

-Plot Specifications Sets are given in a file, specifying:

-The ouput device (HPGL , PostScript, different sizes and orientations), and the output file name.

-The mapping of the layers on the page: superposition/juxtaposition of particular layers.

E. The Operational Application Layer

We have taken the following philosophy for the operational application:

-fixed set of products, with fixed scales: synoptic, meso, micro, Belgium, or diagrams.

-All products are available on the screen, some are automatically drawn on paper, while others can be drawn on request or are denied plotting.

-The contents of the products, as well as the presentation are given by a big specification file, so that the interactive menus are essentially oriented to the product selection.

-The application program and the graphic post-processor are all addressed via a mosaïc giving an overview of all satellite products available.

Conclusions

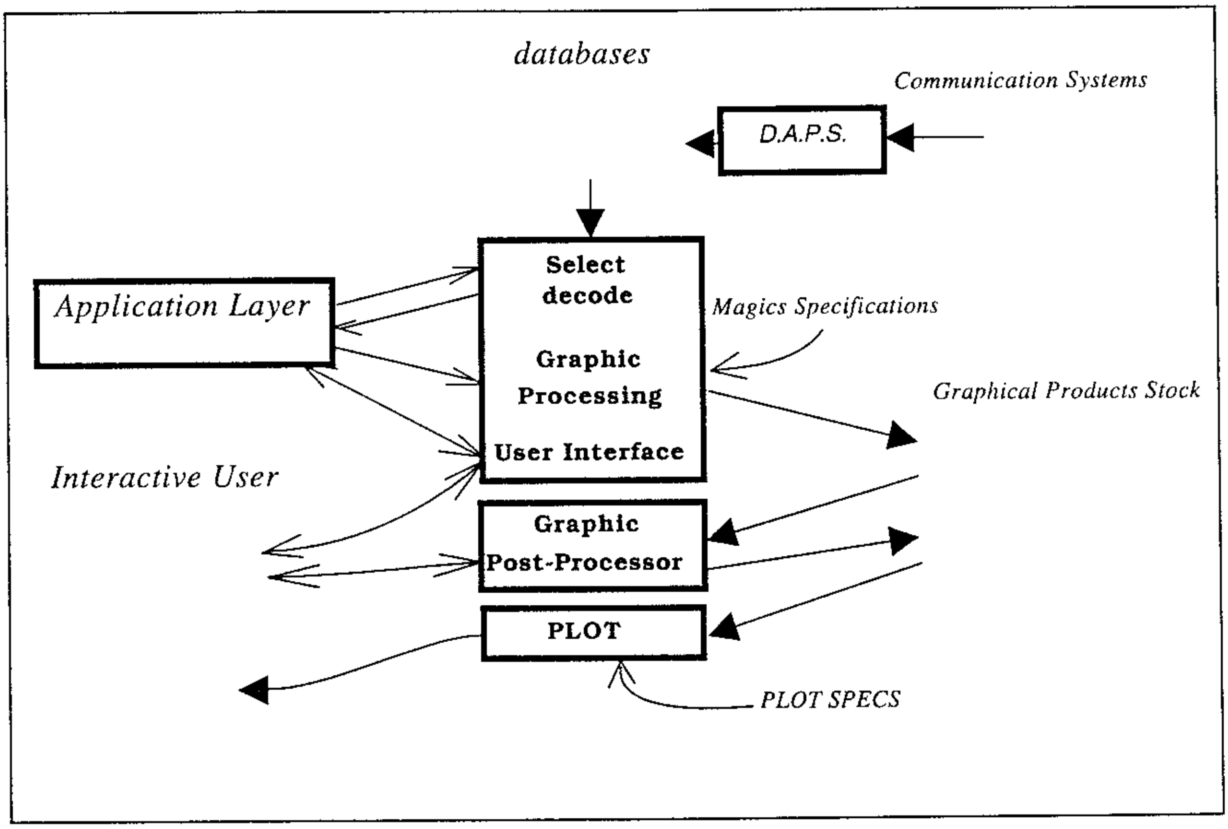
The Belgian Meteorological Service is a rather small institute, localized on one site. The needs for complex databases are not very stringent, and our computing power is too limited to run Global or Meso-scale models.

Nevertheless, we are convinced that a simple specialized system like the one described in this paper is really capable of fulfilling all our needs, without having to develop ourself more complex systems of GUI, which we can leave to the graphics specialists of software firms: this is because the meteorological needs are very specific and well defined.

It is also much more interesting for us to have a locally developed software that we know very good, and that is fitted to our database acquisition system. During the early stages of our developments, we could appreciate the problem of adapting software developed elsewhere, with different computers and data acquisition and representation philosophy.

The care we took to preserve compatibility with the rest of the world: use of the standard GKS without (until now) mixing it externally to other graphic software, use of standard binary decoders and encoders, wide (but not complete) compatibility of ECMWF database data structures with ours..., allows us to be confident as regards the lifetime and the future development of the system.

ANNEX: Conceptual System Layout



Swedish Meteorological and Hydrological Institute

Sweden

RiPP - Project Status and Plans

Background

On the first of July 1992 the SMHI reorganized to become more business oriented. The main reasons were the decreasing grant from the Government and the open competition on the market. To be successful with the reorganization, you also have to change parts of the production system and your way of producing forecasts. The RiPP project was initialized in early 1993 to handle this.

The object with RiPP was to:

- automate the forecast production that can be automated from an economical point of view, not technical

- centralize the production of basic forecasts

- release people from forecast production that can be used for quality increasing supervision, customized production and consultation, product development and market penetration

- no change in forecast quality (at least not lower)

- decrease the cost of forecast production by 20%

Spring 1993 - Result of preliminary study

The preliminary study was finished in April 1993. Several projects had been identified and were scheduled to be carried out from May 1993 to July 1996. During this period SMHI will spend 30 man-years on development of meteorological methods and systems.
Some of the projects were started just before the holiday 1993 and project leaders were assigned to the projects.

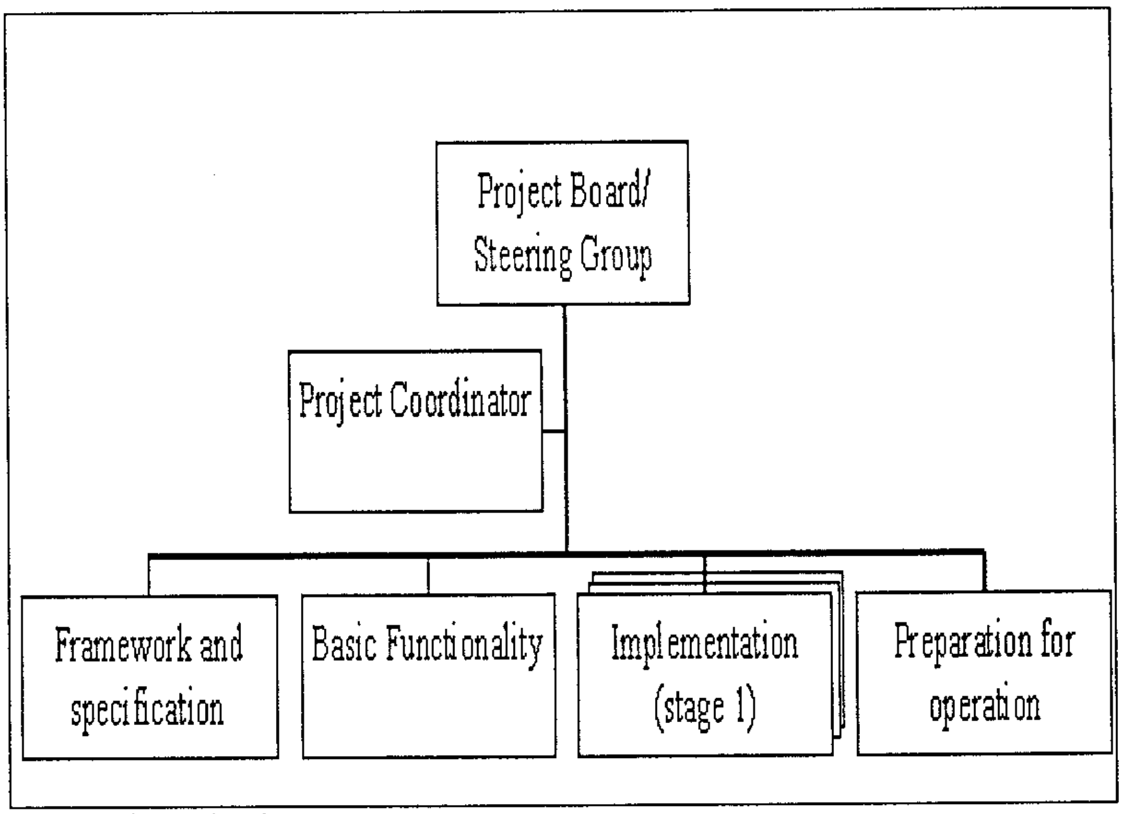
Autumn 1993 - New project organization

RiPPs project board/steering group and project coordinator faced quite soon two major problems:

- The project leaders could not get enough resources to the projects, which delayed the projects.

- It is very difficult to coordinate several projects that run independently over a long time, which are heading for the same goal.

To handle this problem, RiPP was organized according to the picture below.



Project Organization

The project management consists of one common project board/steering group for all projects, to which the project leaders report. The project coordinator works with the daily coordination of the projects on behalf of the project board.

Framework and specifications

This project works with specifications of the technical and meteorological framework. That is defining standards for platforms, development tools, development methods, common API (Application Programming Interface), database system, user interface and so on. The expected result of this project is "the RiPP system architecture" and specification of criterions of an application to fit in the architecture. This will make it easier to by subsystems or system components from external developers. The specification of the architecture will be finished during autumn 1994.

During this first period, the project has worked with an investigation of how NEONS could be used at SMHI (finished in July this year) and has started the work with specification of the system architecture, and investigation of different development tools (GUI builders, and graphical packages).

This project will go on until October 1996. It will of course affect the other projects very much.

Basic functionality

This project has been established in order to find out what functionality the new system should contain. The starting point of this project is the mental process that goes on when creating a forecast. The purpose is to find out how to build systems that support these processes as good as possible. This is done to avoid the usual mistake to only focus on what is technically possible

The project is responsible for defining and verifying methods for:

- supervision of models, observations and forecasts
- validation/verification routines
- graphical production and visualization tools
- automatic production
- statistical interpretations/adaptations

This includes building and testing of prototypes. The expected results of the project are the meteorological requirements on the new tools. Testing of new methods and prototypes is carried out in "the forecast laboratory".

The duration of the project is till the end of the RiPP project.

This far the project has been working with prototypes and requirement specifications for the Implementation Stage 1 and with preliminary study on the forecasters working methodology.

Implementation

This project is responsible for implementing the new tools and systems according to the rules given by the Basic functionality and Framework projects. The implementation will be carried out in five stages according to this schedule:

- Stage 1 operational in October 1994
- Stage 2 April 1995
- Stage 3 October 1995
- Stage 4 April 1996
- Stage 5 October 1996

The date when the result of each stage is operational is not negotiable. Functionality that cannot be implemented on time will be moved to next stage, six months later.

Stage 1 one will be implemented on our present computer environment, VAX/VMS, DECStations/ULTRIX and PC/MS-Windows.

The functionality after stage 1 is:

- Automatically generated three hour precipitation forecasts, based on radar information and HIRLAM forecasts.

-TAF production tool and TAF supervision.

- New visualization tool. This tool can only display radar, satellite and post processed model output. Due to that the user is limited to look at the products, which are generated. There is no cross section or zoom functionality.

- HIRLAM supervision.

Stage 2-5 will be implemented on an environment conforming to the specifications from the Framework and specification project.

Preparation for operation and training

It is very important to prepare the staff to give them the right education and training to successfully make the result of each stage operational. This project takes care of that. Apart from planning and performing the training, the project is also responsible for evaluation of the result of each stage.

Recommendations

Recommendations on Meteorological Workstation development

(as agreed on by EGOWS 5)

1. Requirements for a MWS

- Access to all types of meteorological data: messages, plotted observations, satellite and radar images, fields from NWP, elaborated maps, climatological databases.
- Manipulation of data: interpolation from NWP fields, derivation of additional parameters, trajectories, image manipulation, dynamical or statistical interpretation, verification procedures.
- Visualisation of the above.
- A short response time is important.
- Interactive display of observations and other sources of meteorological information.
- Interactive data manipulation of all products issued from NWP and their display (including cross-sections and meteograms).
- Image manipulation (superimposition, animation, roaming and zooming).
- Facilities to edit maps (adding information and modification of automatically generated information).
- On-the-fly production of as many products as possible instead of using precomputed products.
- Two-dimensional graphical display is the main purpose of operational MWS. The problem of 3D graphics still looks too expensive. Also it should be investigated in which way, it can be put to use in operational forecasting. ECMWF will be testing the product Vis-5D. DWD has tested 3D operationally, but the forecasters did not use it.
- Need to give the forecaster an easy user interface based on icon representation and mouse or equivalent equipment for interactions. Accelerator keys and function keys can be used.
- Modularity of applications with clear seperation between data handling, graphics and user interface is important.

2. Recommendations on standards

Meteorological

- GRIB and BUFR WMO codes for data representation.

  For data, the exchange format should be uniform (following WMO guidelines) but each country should be able to keep its own internal storage format. In particular, for internal handling of observations, an alternative format to BUFR may be considered for performance reasons.
- The extended GRIB format for satellite image data should be considered. The extended GRIB format is being investigated by ECMWF and Brazil. The current recommendation for an extended GRIB has some problems storing tracking information for polar orbiting satellites.
- The use of BUFR for exchange of radar images will be investigated.

Computational

- UNIX operating system, compatible with System V Release 4 is desirable.
- C and C++ programming languages (FORTRAN 90 may be considered). New programming environments such as object oriented programming (OOP) should be considered when designing the next generations of meteorological systems. Tests are being performed in Finland and at ECMWF.
- X Windows System for visualisation using the MOTIF toolkit to develop user interfaces.
- The look and feel of meteorological applications should follow the look and feel of general applications.
- The X Window Systems (Xlib) is the accepted graphics standard. Developments using ISO GKS or PEX/PHIGS might also be considered.
- The ISO Computer Graphics Metafile (CGM) format might be considered especially as an interchange format for meta files.
- Open-GL should be investigated as an upcoming standard.
- For printing purposes, PostScript is a good solution.
- Mac/PC for DTP purposes, eventually commercial DTP-software as an integral part of the MWS. Interactive work based on precomputed maps.

- -

- No recommendation on the hardware platform is given.
- Dessimination of the products to end users: This function might be included in a MWS, but has not been discussed.
- Be carefull about the memory limit on X-terminals.
- The has been reasonably good experiences with Linux on PCs.
- SQL, for accessing orginised databases, should be investigated. Awaiting extended form, which the majority is using anyway.