MAGICS next generation:
an object-oriented architecture with a new contouring package

Sylvie Lamy-Thépaut, ECMWF

Since its first release in 1985, MAGICS has produced an impressive number of meteorological plots. Today we will present how we plan to improve the extendibility of this software by introducing an object-oriented architecture and a more flexible contouring package.

Whilst MAGICS will continue to produce attractive maps, its simpler concepts will allow for greater versatility. The addition of new meteorological data types and formats will be simplified, as will the management of legends, titles and annotations. MAGICS will thus be better enabled to take on the new requirements for a modern piece of software.

MAGICS is a meteorologically oriented piece of graphical software providing a very simple FORTRAN callable user-interface. It is robust software, heavily used at ECMWF to produce meteorological plots and also by some very reputable pieces of software such as Synergie, the French workstation for forecasters, and we should not forget to mention its use in Metview.

Today, we are designing a new MAGICS to meet the new requirements of the meteorological community.

We want to keep and improve the meteorological aspect. Research people nowadays are very productive and are coming every day with new data types. We want to be able to easily integrate their knowledge about this new data in order to produce nice visualisations and meaningful titles.

We also want to be able to integrate new data formats. And we know this is a weak point of MAGICS where internal data representation and external data format are really mixed up. The new design will enable us to add in the next months a PNETCDF action to deal with NetCDF file. The use of the new GribEx will enable us to plot Edition 2 GRIB files, and we are now able to display ODB data.

We will redesign all the legend and title handling. The legend requirements have increased a lot during the last years, and the actual implementation is not satisfactory anymore. The way to specify a legend or a text is confusing. We will simplify the concepts and give the user more controls but also more responsibilities. The result will be more WYSIWYG (What you specify is what you get!)

Metview will be the main winner of this redesign. The communication will now be much simpler (C++ to C++), and the new architecture will take into account the needs of an interactive piece of software. The first consequence will be a major improvement in the legend handling, and soon we will be able to add a lot of new interactions like querying the properties of graphical objects or interactive positioning of the plots.

In this redesign phase, we have to review the requirements to see how they have changed in these last 18 years.

We will introduce new coastlines, trying to adapt the resolution dynamically according to the geographical area. We will offer the possibility to add rivers and towns by implementing a system of layers. We are also working on more sophisticated land shading using topographical information.

We are considering the use of a new contouring package. The problem to solve is still the same: find the right balance between attractive and accurate maps, taking into account the fields increased resolution. A simple linear contouring can give satisfactory results on a global map for a .25 resolution field. And we have to be aware that any complex contouring package will introduce interpolation in the data. Saying that, we would like our contouring tool to be flexible enough to give us the possibility of tuning this interpolation level. This tuning can be done according to the geographical area or to the user needs. Then it becomes more like a tool to inspect the data or understand the weight of the interpolation on the resulting plot.

A second point we would like to address is the ability to work on scattered data or irregular grids.

Our choice is an algorithm called Akima proposed by the Brazilian institute INPE/CPTEC, which, INPE/CPTEC as part of a long cooperation with ECMWF (MicroMagics/Metview), is implementing a C++ version that we are testing now.

We are also considering the new requirements.

The web is at the top of the list, and we have to keep an eye on the possible new formats. We could enable the use of HTML tags in text definitions this will probably simplify the life of users. We received a request to enable the geographical navigation on the produced maps, and we are considering the use of SVG to solve this problem.

Another request is the possibility of adding annotations on maps. Then we will offer the possibility of drawing graphical (like box, or circle) or meteorological objects (like significant weather) on a plot.

To take into consideration all these requirements, we really had to design a solid object-oriented architecture. MAGICS is already object-oriented: of course it is not written in an object oriented language but the principles are here, and looking to its API, it is very easy to make a simple mapping. Each action becomes a class belonging to the same hierarchy, and each parameter becomes an attribute of the class.

From the modern graphical pieces of software, we have taken the concept of Scene Graph, and designed some basic concepts like

- Data Decoders to decode complex data representations (like GRIB, NETCDF) into basic representations
- Visual Algorithms to apply on these basic representations to produce graphical objects
- Drivers to visualise these graphical objects

In this simple skeleton it will be easier to add new data decoders, new visualisation algorithms, or new drivers, or inherit from existing ones.

The Object-Oriented approach comes with nice concepts such as the design pattern that helps in designing difficult notions.

The factory pattern is very useful to address the issue of extendibility. We have a factory of data decoders, projections, and drivers.

The Visitor pattern helps in the legend and title handling. For example a title visitor will visit each plotted data and ask for information for the title, it can use to build the correct title.

C^{++} comes with the STL, which helps in the management of large collections of objects.

Another important issue is how we plan to ensure a smooth migration, and how we see the backward compatibility. Right now, we plan to migrate one action routine at a time, providing we are able to overlay on the same map old and new MAGICS plots. This overlay is quite difficult to achieve. We have created new migration parameters to enable or disable the use of the new action routines when they are introduced, and to disable or enable the migration messages. We will send a lot of migration messages, informing the user about the use of a new action routine, or that he is using deprecated parameters, or that we changed a default value. We also plan to make migration analysis tool that the users can link with their program. This tool will analyse the MAGICS calls and give the user information about possible side effect of the migration.

MAGICS will still produce attractive maps, but the addition of new data types or new types of visualisation will become much easier. The new contouring package will offer more flexibility to users, and they will appreciate the new functionalities such as the drawing of annotations. All these changes will make MAGICS ready for the new needs of the meteorological community.