

Hans-Joachim Koppert ¹
 Deutscher Wetterdienst, Offenbach, Germany

1. INTRODUCTION

The Deutsche Wetterdienst (DWD) develops meteorological workstation systems to support forecast operations at DWD's central and regional forecasting offices. There are 3 different systems assisting guidance and warning operations (Kusch, Koppert, and Pogoda, 1994), interactive chart production and 3D visualization (Koppert and Schröder, 1997). Users are still happy with the current systems, but they became hard to maintain and upgrade. Furthermore they lack requested features like GIS support, advanced analysis functionality, and the seamless integration of 3D visualization and interactive chart modification into one single application.

In 1999 a clear decision was made to replace these aging systems and to move away from dedicated hard- and software environments. The new meteorological workstation project is at the time of writing a joint effort of DWD, MeteoSwiss, and the German Military Geophysical Office. It was decided to develop a new portable system architecture on the basis of the JAVA programming environment.

2. RATIONALE FOR USING JAVA

The current operational workstation systems were built with C, Motif, X11 and OpenGL on the SGI Irix platform. This combination allowed us to deliver very good performance to the user. However, this kind of approach wasn't very portable and we were tied to a single vendor. Porting it to Linux was possible with reasonable effort, but moving to Microsoft Windows would have been too costly.

The new software should not only replace and extend the old one, it should also provide us with the possibility to configure different types of applications, ranging from simple radar displays to full featured meteorological workstations. This also means that we are taking different categories of customers into account. We'll have those with Microsoft Windows based PCs and those with Unix workstations.

Java was chosen as the computing platform, since it promised to meet our requirements, especially the independence from the operating

system and the underlying hardware. The performance has been greatly improved from the early days by new compiler technologies and optimized packages. Within this project it was shown, that even in numerical intensive applications Java can outperform C++. Furthermore Java offers APIs that especially support performant platform independent visualization, like Java2D, Java3D, and Java Advanced Imaging. Its rich set of built-in libraries supports faster implementation time.

Java has already shown its potential in several projects like the VisAD library for building interactive and collaborative visualizations (Hibbard et al., 1999) and FX-Net, the internet client interface to AWIPS (Madine and Wang, 1999). At DWD we have built a prototype meteorological workstation (Koppert et al., 1999) in Java to evaluate the usefulness of Java for this kind of application. Our experiences with this prototype convinced us to start this project.

3. DESIGN GOALS

After collecting, structuring, and prioritizing user requests, the most important requirements have been derived:

- Integrated 2D and 3D visualization (point data incl. surface obs., temps, lightning data; gridded data; satellite and radar imagery; geographical data; bulletins; etc.)
- Interactive and batch chart production
- Data decoding and management
- Client with multiple windows, data layers, animation, analysis functionality/NWP-model monitoring
- Areas of operation: Guidance, warning / consulting, research, climatology, reports, commercial and private customers

The software architecture has to be

- Clear, open and expandable
- Scalable and fault tolerant
- Portable across hardware or operating systems

There are two important design issues: The ability to easily integrate new application layers and to interface with a variety of data sources, like local files, flat file systems or relational databases

¹ *Corresponding author address:* Hans-Joachim Koppert, Deutscher Wetterdienst, Kaiserleistr. 42, 63067 Offenbach, Germany; email: Hans-Joachim.Koppert@dwd.de

4. SOFTWARE ARCHITECTURE

Figure 1 shows the view on the 3 tiers of the new meteorological workstation, called NinJo. Clients can be fat clients or thin clients. Fat clients will be mainly deployed in operational weather forecasting whereas thin clients can serve DWD's staff or external customers.

The server tier provides several server types. There will be an application server to serve thin clients, a batch production which is more or less a headless fat client, and the servers themselves that offer data and computing services.

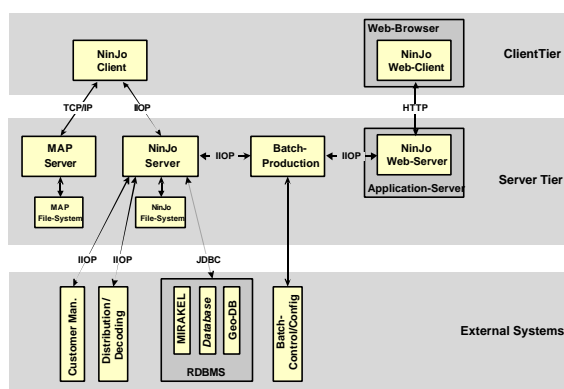


Figure 1: The NinJo architecture

Configurations are stored as XML. Examples of configurations are the clients functionality (i.e. which data layers), bookmarked workspace settings or a batch product description.

4.1 Client

Differently configurable clients are built on a layer framework , where each layer is responsible for one kind of data, e.g. observations, satellite imagery, or gridded data. Each layer can process events and holds its own data. Each layer uses the Model View Controller (MVC) design pattern. A window is made up of several layers, therefore the whole client consists in a hierarchy of MVC triplets.

There is one main window and several secondary windows holding sounding displays, meteograms, text editors etc. The main window (figure 2) looks like the WFO-Advanced Workstation (MacDonald and Wakefield, 1996). After some interactions with our users, this type of window configuration was accepted as the best compromise between static windows like DWD's old legacy software MAP and extensive multi window systems.

4.2 Server

One can see from figure 1 that there are several ways of storing and accessing data. Data

that has to be accessed regularly is stored in flat files on servers at the regional offices, while data that needs only to be addressed sometimes, like climatological or out of area GIS data, will be stored in a RDBMS in DWD's main office.

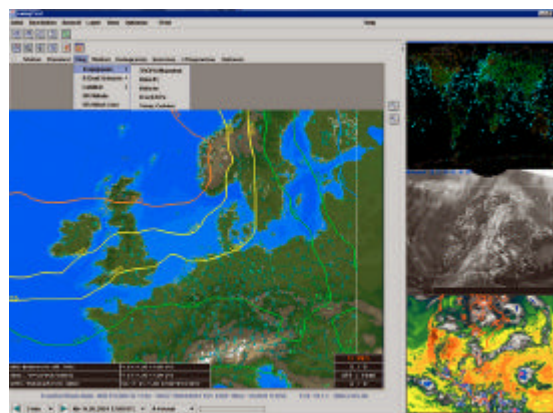


Figure 2: A prototype NinJo GUI

After intensive benchmarking we have chosen CORBA over RMI as middleware. There was neither a significant difference in performance nor in complexity. Therefore CORBA's ability to easily interface with legacy software or additional services, like transactions for secure data storage (e.g. interactively modified model output), turned the balance.

The NinJo data server consists of several data servers of different types. Since we use Corba, it's better to say that there are several CORBA services distributed on a number of server computers. One data server hosts a set of data types with common storage and access characteristics. Examples are point data server (storing BUFR encoded data), gridded data server (storing GRIB encoded data), raster data server (storing unprocessed satellite and radar imagery). All data servers follow one generic design to facilitate the implementation of new data types.

Client side application development is supported by an abstraction layer that makes the client independent from the middleware, it also facilitates load balancing and fail over. On the server side the same layer ("Access layer" in figure 3) abstracts from storage types and allows to transparently access flat files or data bases.

4.3 Visualization

The visualization part of NinJo consists of 3 layers: Graphics Object Factory (GOF), Generic Visualization Library and Meteorological Visualization Library.

The GOF abstracts from the actual graphics API (Java2D, Java3D, OpenGL, SVG, Postscript). It handles the geometry and the

appearance of graphical objects. The GOF is a simplified version of the Java3D scenegraph adopted to the needs of meteorological visualization. Java3D will be used for fast, hardware accelerated 3D (possibly also for 2D rendering), Java2D for diagrams, web graphics, and batch production. Postscript output will be implemented using a Graphics2D-object and the Java Print Service API or by writing a custom driver that traverses the GOF-scenegraph directly.

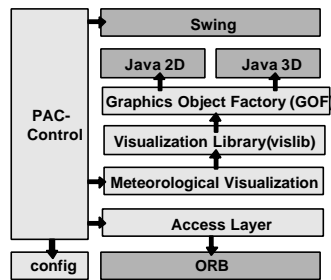


Figure 3: The NinJo client architecture (->: uses)

The generic Visualization Library provides high level graphical tools. Meteorological applications assemble depictions with basic objects like wind barbs, cloud cover symbols, contour lines, and iso-surfaces, only to name a few.

5. THE NINJO PROJECT

The NinJo project is a joint effort of several institutes. It bundles resources that were dedicated to several workstation systems before, because it will offer a superset of the old systems functionalities.

The project started late 1999, the meteorological specification was presented in August 2000, the decision on the technological design was made in may 2001. We expect an evolving prototype that implements all basic functionalities on the server and the client side in spring 2002. After yearly cycles of design, implementation and test we expect rollout mid 2004.

6. REFERENCES

Kusch W., H.-J. Koppert, and M. Pogoda, 1994: The Meteorological Application and Presentation System (MAP) of Deutscher Wetterdienst (DWD), *Tenth International Conference on Interactive Information an Processing Systems for Meteorology, Oceanography, and Hydrology, Nashville, Amer. Meteor. Soc.,200-203*

Koppert, H.-J., Schröder, F, 1997: 3D-Visualization at DWD, *Sixth Workshop on Operational Meteorological Systems, Reading, UK, ECMWF Workshop Proceedings*

MacDonald, A.E and J.S. Wakefiled, 1996: WFO-Advanced: An AWIPS-like prototype forecaster workstation. *Twelfth International Conference on Interactive Information an Processing Systems for Meteorology, Oceanography, and Hydrology, Atlanta GA, Amer. Meteor. Soc., 190-193*

Hibbard, W.S., S. Emerson, C. Rueden, T Rink, D. Glowacki, N. Rasmussen, D. Fulker and J.Anderson, 1999: Collaborative visualization and computation in the earth sciences using VisAD, *Fifteenth International Conference on Interactive Information an Processing Systems for Meteorology, Oceanography, and Hydrology, Dallas, TX Amer. Meteor. Soc., 478-480*

Madine S., N. Wang, 1999: Delivery of meteorological products to an internet client workstation. *Fifteenth International Conference on Interactive Information an Processing Systems for Meteorology, Oceanography, and Hydrology, Dallas, TX, Amer. Meteor. Soc., 356-359*

Koppert, H.-J., H. Haase, O Gehrke, S. Lehmann, 1999: Lessons learned from developing a prototype meteorological workstation in Java, *Seventh Workshop on Operational Meteorological Systems, Reading, UK, ECMWF Workshop Proceedings, 150-155*