

13.11 USING JAVA GRAPHICS TO DISPLAY OCEAN OBSERVATIONS IN NOAA SERVER

Donald W. Denbo*
Joint Institute for the Study of Ocean and Atmosphere,
University of Washington, Seattle, Washington

1. INTRODUCTION

The objective of the NOAA Server project is to develop a framework that will provide a unified view of NOAA's environmental information databases to the Web user. To accomplish these goals NOAA Server provides a Web-based interface that allows the user to search, preview, and download NOAA's data holdings.

A Web-based Java client presents a user interface that allow the user to select and display subsets from NOAA's data library (Brazille et al., 1997). The scientific graphics toolkit (sgt) was developed to provide a NOAA Server user interactive preview and plot overlay capabilities.

2. SCIENTIFIC GRAPHICS TOOLKIT

The scientific graphics toolkit (sgt) provides the basic building blocks for developing graphic applications and applets in Java.

2.1 Design Goals

Sgt was developed primarily to provide client-side graphics for the NOAA Server project (Denbo, 1997). Existing java graphics class libraries were considered, but were not used since they did not support one or more of the design goals. Sgt was designed to support the NOAA Server project by:

- Allowing a graphics client developer a great deal of flexibility and freedom.
- GIS style layer approach to display geophysical data
- Support several types of graphical display
 - X-Y plot
 - 2-D contour and "pixel" plots.
 - Vector plots.
 - Point-Value plots.
- Develop a framework that is easily extended

2.2 Implementation

Sgt uses two separate coordinate systems (physical and user) and translates these to device coordinates. Device coordinates are the JDK1.1 graphics coordinates (pixels) where the origin is in the upper left. Physical coordinates have a linear transformation to device coordinates and are floating point quantities with the origin at the lower left. For printing, the physical coordinates

(either inches or millimeters) determine the actual printed position and size of sgt graphics. The transformation of user coordinates to physical coordinates is arbitrary and defined by the developer.

Sgt has three main components (Figure 1) that reflect

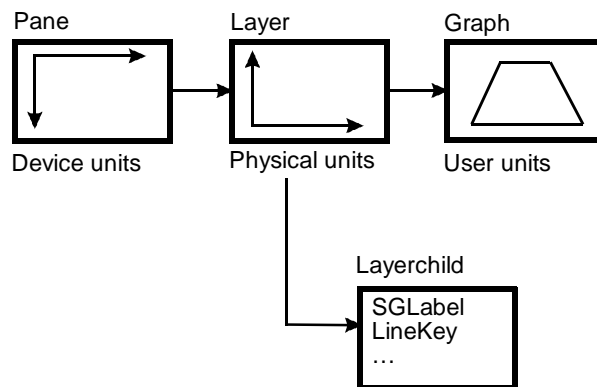


Figure 1. Main sgt graphical components.

these coordinate systems. These components are: the Pane, on which all graphics are drawn; the Layer, which insulates the developer from the device coordinates of the pane; and the Graph, which provides the transforms from user coordinates to physical coordinates.

Pane. The Pane can either draw on the screen, using double buffered images, or on a user supplied Graphics object. Sgt determines whether the supplied Graphics object is associated with a printer or off-screen image to properly scale the output. The Pane also manages the mouse events and drawing the associated Layers. Many Layers can be associated with a single Pane.

Layer. Each Layer can have at most a single Graph. However, any number of LayerChild objects can be associated with each Layer. LayerChild objects specify their size and position on the Layer in physical coordinates (thus they are independent of any user transformations that are active for the Graph). SGLabel, LineKey, ColorKey, and Logo objects are some of the LayerChild objects available.

Graph. CartesianGraph, MapGraph, and PolarGraph are examples of Graph classes that can be associated with a Layer. (Only the CartesianGraph is currently implemented.) While the CartesianGraph must specify x and y transformation objects, it can have any number of x and y axes. The appropriate Renderer object, responsible for the graphical representation of the data, is auto-

* Corresponding author address: Donald W. Denbo, NOAA/PMEL/OCRD, 7600 Sand Point Way NE, Seattle, WA 98115; e-mail: dwd@pmel.noaa.gov

matically chosen when the data and rendering attribute objects are associated with a CartesianGraph.

2.3 Mouse Events

Mouse events are used to notify the application that an object has been selected, a zoom rectangle is available, or that an object has been moved. Since multiple Layers can exist it is not always clear to which Layer a zoom operation should be applied. Thus, sgt makes the zoom rectangle available, but leaves the operation of zooming to the application.

3. NOAASERVER APPLICATION

The NOAAserver client allows the user to select a remote dataset, navigate through the dataset, request a small data subset from the dataset, receive the data request, and graphically display the data. The graphical display is accomplished using sgt. Sgt, however, is a low-level graphical toolkit and an additional software layer is required to use sgt.

3.1 Overview

The approach used here is to embed the sgt graphics into an independent window, the PlotFrame, which descends from java.awt.Frame. Plot Frame is a container that contains a menu bar, button bar, and a GraphicLayout object (Figure 2). Several classes descend from

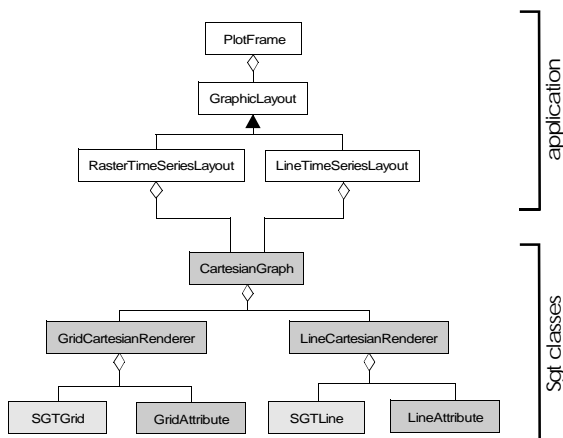


Figure 2. UML class description for principle classes in NOAAserver applet. The shaded classes are members of sgt.

GraphicLayout, including RasterTimeSeriesLayout, LineTimeSeriesLayout, and StationPlotLayout. The GraphicLayouts are responsible for managing the main graphics Pane, key Pane, CartesianGraph, transforms, axes, and labels. The GraphicLayouts also manage object selection and zooming.

3.2 Station Locations

The StationPlotLayout uses the LineCartesianRenderer with SGTLine and LineAttribute objects. The LineAttribute is set to specify the plot mark rendering option. In Figure 3, two sets of station locations are plotted using

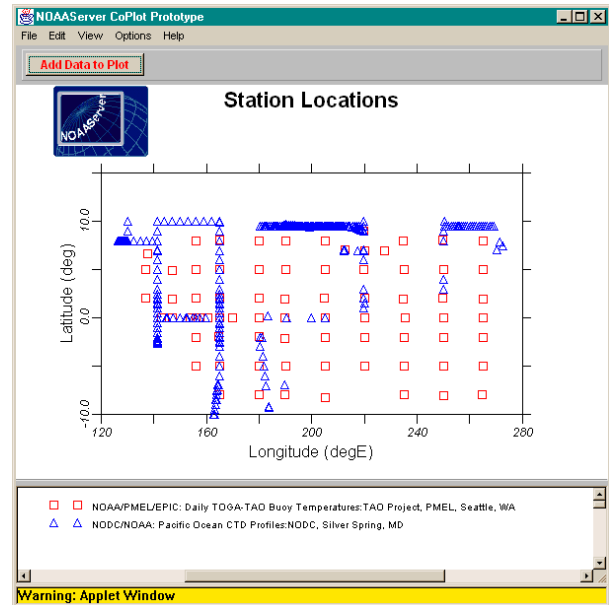


Figure 3. Station locations for the TOGA-TAO moorings and selected CTD observations plotted using StationPlotLayout.

two Layers. The first Layer includes the SGLabel, Logo, and a CartesianGraph. The first CartesianGraph includes the axes and data that represents the TOGA-TAO mooring locations. The second Layer only contains a CartesianGraph which in turn only includes the data for the CTD measurement locations.

3.3 Time Series Plot

The LineTimeSeriesLayout also uses the LineCartesianRenderer, but with a LineAttribute object that specifies the solid line rendering option (Figure 4). In this example, two data subsets from different remote servers share a single set of cartesian transforms and axes. The two data subsets are plotted using two Layers, with the first containing the shared graphical objects and the first data subset, and the second containing only the second data subset.

Since time can be a user coordinate, the graphical display of a time series is accomplished by creating a transform that maps time to physical coordinates and using a TimeAxis. The TimeAxis class automatically uses one of five TimeAxisStyles, from YearDecadeAxis to Minute-HourAxis, to display time. These styles take into account the number of minutes in an hour, hours in a day, leap years, etc., to place tic marks and labels on the axis.

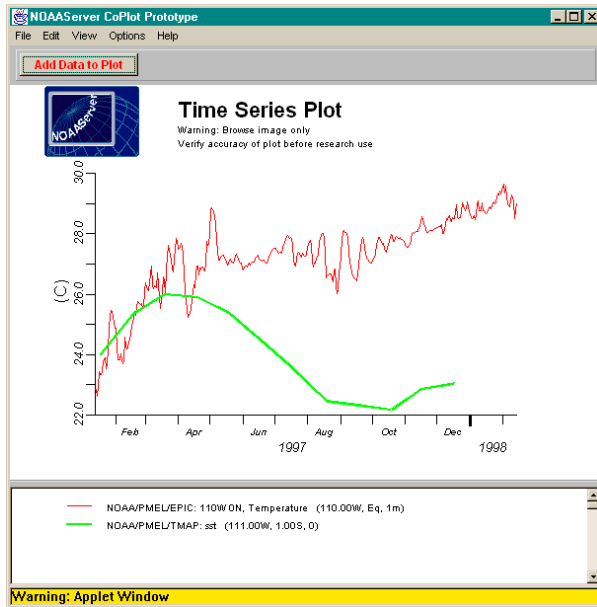


Figure 4. Time series plot of sea surface temperature (SST) from January 1997 through February 1998 using LineTimeSeriesLayout. Measurements from the TAO array at 110W on the equator are co-plotted with the COADS climatology.

3.4 Raster Time Series Plot

The RasterTimeSeriesLayout uses the RasterCartesianRenderer with SGTGrid and GridAttribute objects. The GridAttribute includes a ColorMap and a transform to specify how data values are mapped to colors and specifies the raster rendering option (Figure 5). The RasterTimeSeriesLayout is designed to render a single data subset.

4. FUTURE DIRECTIONS

The second version of sgt is presently under development. Lessons learned during the implementation of the NOAA Server client graphics are being used to reorganize the sgt classes (Denbo, 1998). The new class organization will make sgt easier for a developer to use and also provide additional flexibility.

Presently sgt is implemented using Java JDK1.1.x. JDK1.1.x imposes many restrictions on the graphical developer. For example, line width, dash styles, and label orientation, must all be implemented by the developer. JDK1.2 promises to alleviate these limitations when finally made available. The addition of line rendering styles and a much more flexible string rendering mechanism should significantly improve sgt's capabilities.

The MapGraph class and its supporting classes need to be designed and implemented. These classes will provide developers to display geographical data in a wide number of map projections.

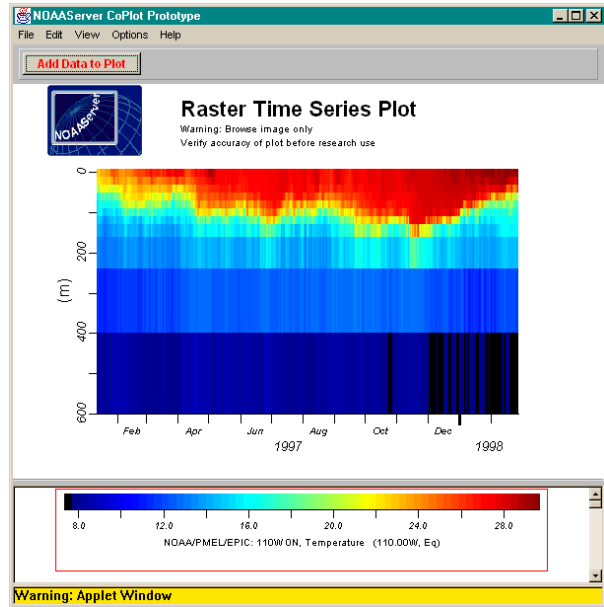


Figure 5. Time-depth raster plot of temperature from January 1997 through February 1998 at 110W on the equator using RasterTimeSeriesLayout.

5. ACKNOWLEDGEMENTS

This work was supported under NOAA Cooperative Agreement NA67RJ0155 through the Joint Institute for the Study of the Atmosphere and Ocean (J. I. S. A. O.), University of Washington. This publication is JISAO contribution number 596. The views expressed herein are those of the author(s) and do not necessarily reflect the views of NOAA or any of its subagencies.

6. REFERENCES

- Brazile, W., D.W. Denbo, and W.H. Zhu, 1997. NOAA Server Version 2 Co-plotting Prototype. Presented at *NOAA WebShop97*, October 22-23, 1997, Silver Spring, Maryland.
- Denbo, D.W., 1998. Scigraph: Object-Oriented 2D Scientific Graphics Library. Presented at the *14th International Conference on Interactive Information and Processing Systems for Meteorology, Oceanography, and Hydrology*, January 12-16, 1998, Phoenix, Arizona.
- Denbo, D.W., 1997. NOAA Server Graphics Engine Architecture. Presented at *NOAA WebShop97*, October 22-23, 1997, Silver Spring, Maryland.