

Donald W. Denbo*

Joint Institute for the Study of Ocean and Atmosphere, University of Washington, Seattle, WA

1. INTRODUCTION

The Scientific Graphics Toolkit (SGT), a library of java graphics classes, facilitates the development of platform independent Java applications and web applets to produce highly interactive, flexible, publication quality, object-oriented graphics of scientific data. Features include user settable or automatically scaled axes, sophisticated, automatically self-scaling time axes, mouse-movable objects (labels, line keys, icons), customizable objects, automatic generation of legends to explain the data being displayed and contour plots.

SGT, originally developed for NOAA's NOAA Server project (Daddio et al., 1999; Soreide and Daddio, 1998), has been used in the development of OceanShare, a collaborative tool for in-situ oceanographic data sets, and ncBrowse, a graphical netCDF file browser. For convenience of use, SGT conforms to the JavaBeans component architecture guidelines and can be used with both AWT and swing projects.

2. SCIENTIFIC GRAPHICS TOOLKIT

2.1 Design Goals

SGT was designed to support scientific graphics on the client 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 Architecture

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 transfor-

mation of user coordinates to physical coordinates is arbitrary and defined by the developer.

SGT has three main components (Figure 1) that reflect

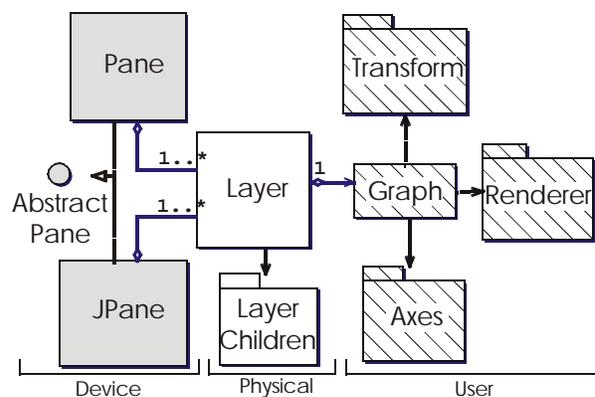


Figure 1. Main SGT graphical components.

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 and JPane, for AWT and Swing projects, respectively, 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 automatically chosen when the data and rendering attribute objects are associated with a CartesianGraph.

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

Axes. SGT provides both space and time axes for use with a CartesianGraph. The SpaceAxis provides a visual scale for the spatial user to physical coordinate transform and TimeAxis creates a visual scale for the GeoDate to physical transform. The GeoDate class extends java.util.Date, providing methods for subtracting times, computing offsets, and creating formatted strings. GeoDate *always* use the UTC (GMT) time zone for parsing and formatting. TimeAxis has several “styles” that can be used depending on the temporal span. If the TimeAxis.AUTO option is used then TimeAxis determines which style to use from the temporal range. Changing the time range interactively (zooming) will automatically change the style!

2.3 Data Model

The SGT data model consists of data type interfaces, coordinate system interfaces, container classes, and a metadata class (Figure 2). SGT has interfaces for point,

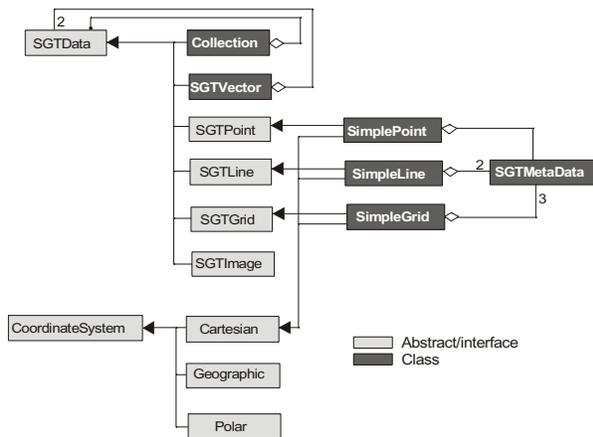


Figure 2. SGT data model classes.

line, grid and image data. These data types were implemented as interfaces to provide the greatest flexibility for developers using SGT. Developers can either implement the appropriate data type interface in an existing class that already provides data or use one of the concrete classes provided by SGT (SimplePoint, SimpleLine, or SimpleGrid). The interfaces extending CoordinateSystem (Cartesian, Geographic, and Polar) do not define any methods that need to be implemented, however, the interfaces are used to indicate the coordinate system to SGT. Classes that can contain other data types, which can include point, line, grid, or image data, are the Collection and SGTVector classes. Finally, the SGTMetaData class encapsulates information about variables or axes, e.g., name, units, etc.

2.4 Events

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 coordinates of the zoom rectangle available, but leaves the implementation of zooming to the application developer.

Property Change Events. Attribute, labels, and color map classes generate property change events when interactively moved on screen or a property is changed. These classes are automatically registered with Pane and JPanel when added to a Layer or Graph. Pane and JPanel are then notified of changes and can then regenerate the graph. This automatic regeneration, which can be temporarily defeated, frees the application from the bookkeeping and logic needed to determine when to redraw the graph. These events can also be used to notify an application if a property has been modified. For example, the notification can be used to inform a dialog that a property has changed and a widget may need to be updated.

3. APPLICATIONS AND EXAMPLE

3.1 Collaborative Tool for in situ Data Visualization

OceanShare was developed to allow scientists to effectively communicate about data and analyses in a collaborative environment. OceanShare uses network data access developed for the NOAA Server prototype data browser and collaborative capabilities provided by NCSA's Habanero framework. (Information about obtaining and installing OceanShare can be found at "<http://www.epic.noaa.gov/collab>". Habanero is available at "<http://www.ncsa.uiuc.edu/SDG/Software/Habanero>".)

OceanShare has been implemented employing LineProfileLayout which uses the LineCartesianRenderer with SGTLine and LineAttribute objects. The LineAttribute is used to specify the line style for plotting (highlight, mark, or solid) and color. In Figure 3 five profiles are plotted using five separate Layers. The first Layer includes the SGLabels and a CartesianGraph. The first CartesianGraph includes the axes and the first profile data. Subsequent Layers contain a single CartesianGraph and profile data set. The LineKey contains key metadata for the profile (latitude, longitude, time) and selecting a key entry allows the user to highlight a profile.

3.2 ncBrowse: netCDF Interactive File Browser.

ncBrowse enables a user to view the contents of a netCDF file using the JPlotLayout class from SGT. JPlotLayout is a flexible class that uses a SGTData object to determine whether to use time or space axis for the x or y coordinate. JPlotLayout also supports both line and grid (1 or 2 dimension) data, creating a LineKey or ColorKey, respectively. JPlotLayout supports interactive zooming and object selection.

Relative humidity from the Vegetation/Ecosystem Modeling and Analysis Project (Kittel et al., 1997) is shown in Figure 4. JPlotLayout determines that the data is of type SGTGrid and doesn't have a time axis. The grid renderer skips any missing data, indicated by Not-a-Number (NaN), in the data arrays.

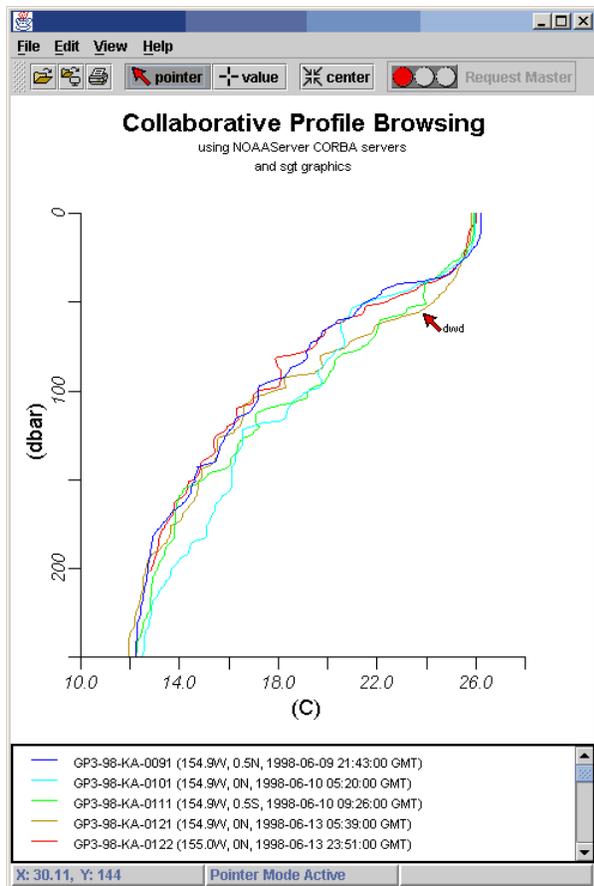


Figure 3. Temperature measurements from CTD observations near 0N 155W.

The second example comes from the OLEM numerical model (Denbo and Skillingstad, 1995). A time series of horizontally averaged surface Kinetic Energy is extracted from the netCDF file and packaged as a SGTLine object (Figure 5). JPlotLayout recognizes that the data is a line with an temporal x axis. The LineAttribute was interactively edited to set the style = "MARK" and the graph was "zoomed" to a time near the onset of the strong winds. The LineKey was used by ncBrowse to indicate the domain of the displayed data.

3.3 TAOMap: Point Data Demonstration.

The TAOMap application was constructed from basic SGT components. The application reads two data files that contain the latitude, longitude, and a label for TAO and TRITON buoys (McPhaden et al, 1998) (Figure 6). A SimplePoint object is created for each line in the data file and added to a Collection object for each file. Each Collection is then added to a separate Graph object with a PointAttribute object. The PointAttribute object has properties for plot mark, mark size and color, and label size, color and position.

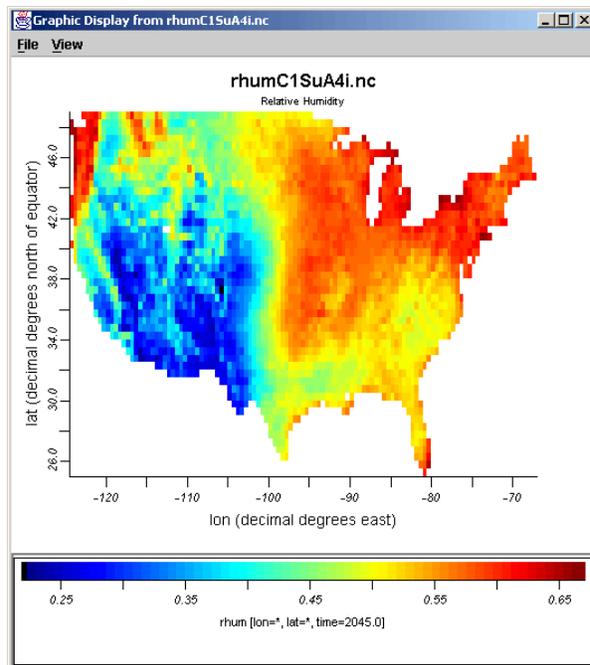


Figure 4. Relative humidity from VEMAP Phase II for model year 2045.

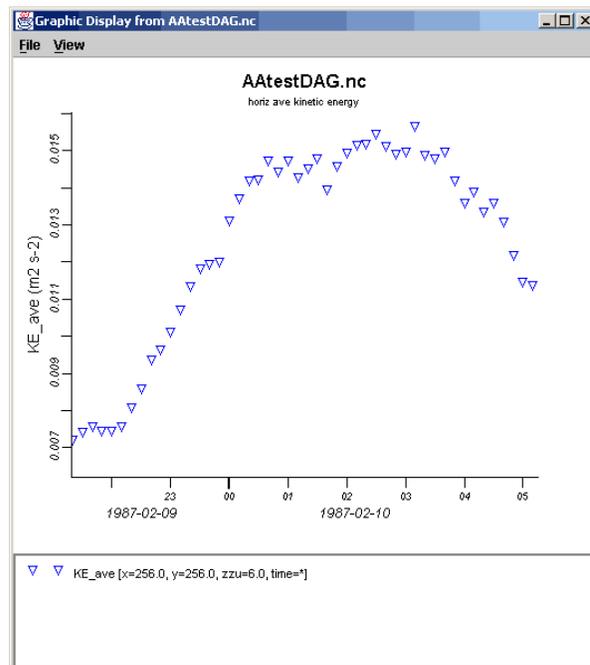


Figure 5. Time series of horizontally averaged Kinetic Energy from an OLEM simulation of a strong wind event in the Gulf of Maine during Feb. 9-10, 1987.

4. AVAILABILITY

This software is provided by NOAA for full, free and open release and is available at http://www.epic.noaa.gov/java/sgt/sgt_download.shtml. More

