Introduction to Chaco

Overview

Chaco is a *plotting application toolkit*. It is meant to facilitate writing plotting applications of all levels of complexity, from simple scripts with hard-coded data to large plotting programs with complex data interrelationships and a multitude of interactive tools. While Chaco can be used to generate static plots for publication and presentation purposes, its components are designed to be used for interactive data visualization.

Basic Concepts

There are a few basic classes of objects that are used to build Chaco plots.

DataSource

Data sources are Chaco's wrapper objects for the actual data that it will be handling. They provide methods for retrieving data, estimating a size of the dataset, indications about the dimensionality of the data, a place for metadata (such as selections and annotations), and events that fire when the data gets changed. There are two primary reasons for the datasource class: it provides a way for different plotting objects to reference the same data, and it defines the interface for embedding Chaco into an existing application. In most cases, the standard ArrayDataSource will suffice.

Interface: AbstractDataSource
Subclasses: ArrayDataSource, PointDataSource, ImageDataSource

DataRange

A DataRange expresses bounds on data space of some dimensionality. The simplest data range is just a set of two scalars representing (low, high) bounds in 1-D. One of the important aspects of DataRanges is that their bounds can be set to "auto", which means that they automatically scale to fit their associated datasources. (Each DataSource can be associated with multiple ranges, and each DataRange can be associated with multiple datasources.)

Interface: AbstractDataRange
Subclasses: DataRange, DataRange2D

Mapper

Mappers perform the job of mapping a data space region to screen space, and vice versa. There are two kinds of mappers in Chaco: LinearMapper and LogMapper.

PlotComponent

All visual components in Chaco subclass from PlotComponent. It defines all of the common visual attributes like background color, border styles and color, and whether the component is visible. (Actually, most of these visual attributes are inherited from the Enable drawing framework.) More importantly, it provides the base behaviors for participating in layout, handling event dispatch to tools and overlays, and drawing various layers in the correct order. Subclasses almost never need to override or customize these base behaviors, but if they do, there are several easy extension points.

PlotContainer

PlotContainers are Chaco's way for handling layout. Because they logically partition the screen space, they also serve as a way for efficient event dispatch. They are very similar to sizers or layout grids in GUI toolkits like WX. Containers are subclasses of PlotComponent, thus allowing them to be nested. BasePlotContainer implements the logic to correctly render and dispatch events to sub-components, while its subclasses implement the different layout calculations. Chaco currently has three types of containers.

Interface: BasePlotContainer
Subclasses: OverlayPlotContainer, HPlotContainer, VPlotContainer