

Overview

Introduction

Bard is an open source application built upon the Eclipse Rich Client Platform (RCP) that provides core components from which new applications can be built. Bard was developed alongside the Analysis Framework and together form a modeling environment where multiple scientific disciplines can work together to understand interdependencies, validate results, and present findings in a unified manner. Bard provides an extensible software platform that helps bridge the time-from-discovery gap among researchers, practitioners and decision makers.

Technologies

Bard uses a number of externally and internally developed libraries to provide window/widget management, data/metadata management, GIS functionality, 2D & 3D visualizations, charting and reporting. These libraries include Eclipse RCP, Tupelo, GeoTools, Visualization Toolkit (VTK), JFreeChart, and KTable. Bard represents a layer of abstraction that exposes the functionality of these libraries through a common API that other projects can use as the basis for new applications. Many developers that use Bard can simply focus on adding the science for their domain specific plugins without worry about how the underlying pieces work (e.g. workflow, annotation, importing shapefiles, etc).

Eclipse Rich Client Platform (RCP)

[Eclipse RCP](#) is a open source platform for building and deploying rich client applications. It includes Equinox, a component framework based on the OSGI standard, the ability to deploy native GUI applications to a variety of desktop operating systems such as Windows, Linux and Mac OSX and an integrated updated mechanism for deploying these desktop applications from a central server. The extensible, plug-in based architecture enables component development to be shared across multiple communities and create cross-platform software solutions that are robust and well supported in both the open and enterprise software communities.

Tupelo

[Tupelo](#) is a semantic content library developed at NCSA that provides an important layer of abstraction above the diverse RDF software ecosystem. It manages both the data and metadata imported into the Bard and provides the ability to search, tag, and annotate the imported data. Tupelo allows Bard to track the provenance of data, a type of virtual paper-trail so that users can determine information such as which algorithms were used, the date it was created, the author, what machine was used, etc.

Geotools

[Geotools](#) is an open source (LGPL) Java library which provides standards compliant methods for the manipulation of geospatial data. The Geotools library implements [Open Geospatial Consortium \(OGC\) specifications](#) as they are developed, in close collaboration with the [GeoAPI](#) project. GeoTools provides the GIS functionality for Bard as well as some of the visualization capabilities (e.g. 2D Rendering).

Visualization Toolkit (VTK)

[Visualization Toolkit \(VTK\)](#) is an open source, cross platform library for 3D computer graphics, image processing and visualization. VTK consists of a c++ class library and several interpreted interface layers including Java, Python and Tcl/Tk. VTK provides the 3D visualization capabilities in Bard since Geotools currently only supports 2D representations.

JFreeChart

[JFreeChart](#) is an open source (LGPL) Java chart library that makes it easy to display professional quality charts. JFreeChart supports a wide range of chart types and is easy to extend if something is not supported. The charting in Bard utilizes both JFreeChart for creating charts as well as some in-house extensions that use the underlying data structure from JFreeChart.

KTable, NatTable and Glazed Lists

The initial table implementation in Bard was provided by [KTable](#), an open source custom SWT Widget that provides a flexible grid of cells to display data. KTable was chosen because it provided more flexibility than the native SWT Table control. Recently, we have experimented with [NatTable](#) and [Glazed Lists](#) since KTable is no longer under development and NatTable is very good with large datasets. Glazed Lists is a more robust implementation of the List API and includes support for dynamic filtering, live sorting, and concurrency. This means that the tables will update as the input list changes so the table is always up-to-date.