Introduction to the Eclipse Advanced Visualization Project

Robert Smith, Scientific Software Development Team, Oak Ridge National Laboratory
Outline

• Introduction to Eclipse
• Motivations for scientific visualization in Eclipse.
• Example visualization services
  – Graph plotting
  – VisIt and ParaView
  – Geometry and Mesh editing with JavaFX
• Setting up remote connections
• Future Developments
What is Eclipse?

• An Integrated Development Environment (IDE).
• A foundation.
• A community.
• A place for Research Software Engineers
What is the Eclipse Foundation?

• Managed by the Eclipse foundation.
• Not-for-profit corporation that maintains the Eclipse ecosystem
• Members include ORNL and other industry leaders like IBM, Oracle, and Red Hat
• Projects are official initiatives of open source development to benefit the community.
• Working Groups collaborate on focused development.
What is the Eclipse Workbench?

- Uses the Open Service Gateway Initiative (OSGi) to manage the modular packages that make up the application.
- Provides the Rich Client Platform (RCP) as a way to extend the workbench to create custom apps by writing new OSGi plugins.
- Open source.
- Written in Java.
Motivation
Motivation for Integrating Visualization into a Workbench

• Simulation workflows span from writing code to visualizing results.

• Having a single RCP app which centralizes and simplifies all these actions would provide a useful tool for scientists.
  – Provides a convenient place for performing all parts of the workflow
  – Allows for use case specific knowledge to aid in workflow design (for e.g. error checking).
Motivation for Integrating Visualization into a Workbench

• ...But other, non-traditional use cases exist for which the ability to visualize files would be useful.

• The Eclipse Advanced Visualization Project (EAVP) was created to provide a framework of visualization resources for diverse RCPs.

• It has been evaluated by:
  – Marintek for creating oceanographic geometry.
  – Bosch evaluated it for use in visualization of automobile parts.
  – As well as Diamond, Itema, and Airbus
EAVP Overview
What is EAVP?

• EAVP features a range of services for different kinds of visualization use cases.
• Provides a flexible API for integrating visualizations.
• Native visualizations in Eclipse SWT and embedded JavaFX.
• Third party visualizations.
• Published as a p2 repository to be drawn into RCP applications.
Visualization Service Architecture

• Each type of visualization is implemented as its own OSGi bundle.

• Bundles contain an implementation of IVizService that can draw the visualization to a composite.

• The RCP app can select which visualization capabilities it wants by finding the right IVizService.

• Bundles may also contain extensions to the preferences menu to allow the user to configure the service.
Visualization Service Architecture

- `IVizServiceFactory`
  - `getServiceNames()`
  - `getService()`

- `IVizService`
  - `createPlot()`

- `AbstractVizService`

- `ConnectionVizService`

- `ParaViewVizService`

- `IPlot`
  - `draw()`

- `AbstractPlot`

- `ConnectionPlot`

- `ParaViewPlot`
Leveraging EAVP Services to Visualize Files in the Workspace

• Visualizing a file is as simple as handing it to the correct service
• Each IVizService provides a list of compatible file types.
• Invoking createCanvas() or createPlot() returns an IVizCanvas or IPlot.
• Using .draw() will draw it to a composite.
Setting Up Connections to Third Party Software

Connections to local or remote machines are supported.
Setting Up Connections to Third Party Software

• Preference menu contributions configure connection
• Uses the parallel tool platform’s connections as a base.
• Can be configured to open connections automatically on startup.

Preferences Menu in Workbench
Examples - 2D Plotting
2D Plotting

• The simplest example of a visualization service is the CSV graph implemented with SWT-XY-Graph.
• The service takes a .csv or .dat file as input.
• Parses the file to try to guess the delimiter.
• Draws a plot containing the file’s data to the given composite.
• Has controls for editing how the graph is displayed.
• Also includes the data from the file in a text editor.
2D Plotting
Examples - VisIt and Paraview
VisIt and ParaView Integration

- VisIt and ParaView are two of the most popular visualization programs for scientific data.
- Open source.
- Scale to visualizing even HPC data.
- Handle a large variety of file types, including .exo, .gen, .nemesis, .silo, .nek5000, and .xyz

Visualizations in an Eclipse workbench using ParaView (top) and VisIt (right).
VisIt and ParaView Integration

- VisIt/ParaView must be downloaded and installed separately from Eclipse, not necessarily on the same machine.
- The EAVP service allows the user to configure a connection to the third party program.
- When the connection is opened, EAVP will launch VisIt/ParaView in the background.
VisIt and ParaView Integration

- When EAVP is set to visualize a file, it will pass the file to VisIt/ParaView.
- The visualization program will open the file and create a model as normal. Images are sent back to Eclipse to paint to the screen.

Temperature map for a tristructural-Isotropic fuel pellet, visualized in VisIt.
VisIt and ParaView Integration

- Interacting with the canvas (eg by clicking and dragging) sends commands to program.
- Other functionality (such as setting the model type or exposing Python scripting) done through UI.

A crash test dummy model in VisIt.
Examples - 3D Modeling in Java FX
3D Modeling with JavaFX

• JavaFX allows for displaying a 3D graphics scene.
  – Shipped as part of the JDK.
  – Lacks potential copyright problems from e.g. LWJGL.

• Intended for casual development without advanced graphics features (eg direct shader support, access to matrix transforms, custom vertex stream configuration, etc.)

• Better suited to real time editing than heavy duty visualizers.
3D Modeling with JavaFX: Technical Concerns

- e(fx)clipse project integrates it with Eclipse.
- A JavaFX scene is embedded directly into a Eclipse SWT composite.
- User events are transparently forwarded to scene.

A fluid impeller modeled in the Geometry Editor.
3D Modeling with JavaFX: Mesh Editor

• Mesh Editor allows for editing of 2D meshes.
• User may create new polygons by clicking.
• Existing meshes can be edited either by mouse or through the properties view.
• Properties may be assigned/edited for polygons/lines/points.
• Currently limited to fluid dynamics boundary conditions
3D Modeling with JavaFX: Mesh Editor
3D Modeling with JavaFX: Geometry Editor

- Geometry Editor is an editor for 3D structures.
- Uses constructive solid geometry.
- Provides basic sample shapes, but main use is importing geometry files.

Shrinker, drill bit, and fastener in the geometry editor
3D Modeling with JavaFX: Geometry Editor

- Supports a variety of file types (.stl, .obj, .vtk, .mtl, .iges)
- Files are read through Xtext DSLs provided through Eclipse extension points.
- Related data structures for file reading have been moved to the January Eclipse project.

A teapot obj file

A hard drive stl file

"VTK" spelled out by a vtk file
3D Modeling with JavaFX: Geometry Editor
3D Modeling with JavaFX: Geometry Editor

Data for shapes is kept in properties view.
Future Developments

• Integration with the Eclipse Advanced Scripting Environment (EASE) to allow interoperability between the visualizations and Python scripting.

• Expand to cover use cases outside of Eclipse RCP, with a focus on working on the web.
  – Vaadin framework
  – Swing
  – JavaFX
More Future Work...

• Offering alternate graphical engine implementations for services.
• In Situ visualization support.
• Full 3D mesh editing and better finite element support.
• Paper under development
Questions?

• Thanks to…

  – Our research sponsors.
  
  – My colleagues at ORNL who have contributed to EAVP, including Jay Jay Billings, Alex McCaskey, Greg Watson, and Anara Kozhokanova
  
  – Sandia National Laboratory for collaboration with the Sandia Analysis Workbench (SAW).
  
  – Chemclipse for providing a SWTChart visualization service.
  
  – L33t Labs for collaboration on JavaFX Geometry Editor.
Links and Getting Started

EAVP Project Page
http://projects.eclipse.org/projects/science.eavp

GitHub Repo
https://github.com/eclipse/eavp

OSGi Beginner’s Tutorial for consuming visualization services
https://wiki.eclipse.org/EAVP_Service_Integration