VERAView

1 Overview

VERAView's purpose is to provide an extensible, easy-to-use viewer, browser, and analysis capability for VERA output. It reads HDF5 files that adhere to the VERAOUT specification and with support for some extensions to the spec. [1] Datasets are categorized as channel, detector, pin, or scalar data. With the exception of the detector category, the dataset shape determines its type.

VERAView is architected around a data model, an event system representing user selections, and a widget framework. The data model represents the content of a VERA output file and provides the necessary processing to categorize and access the datasets it contains. Events represent the "state" of the user's current selections, such as the state point, axial level, assembly, pin, and channel cell.

Ultimately, VERAView is a graphical user interface (GUI) tool implemented as a container of user-selectable widgets, where each widget is responsible for providing a view or representation of some portion of the data. Individual widgets may process whatever datasets are of interest, but are typically focused on a particular type or category. Although widgets in the initial distributed prototype consist of 2D views and plots, 3D views can and will be added in the future. All widgets receive events representing the user selection state, and each widget can be the source of such events.

Several widgets have been implemented in the initially distributed VERAView prototype. These are described below.

2 Requirements

VERAView's features are a result of requirements derived from the rapid prototyping process by which it has been and continues to be developed. Of the many specific functional requirements that have been captured, a small subset has been implement in the initial prototype version. Described here are the high level requirements motivating the existence of VERAView.

VERA-specific tool.

Unfortunately, general-purpose data analysis and visualization tools can be difficult to adapt to a specific use case or operational concept. Thus, first and foremost VERAView is to be a VERA-specific tool, aimed specifically at processing VERA output.

Easy to use.

Applications such as ParaView and VisIt are powerful, general-purpose data analsysis andvisualization tools. [2], [3] Setting up data sources and rendering pipelines requires a fair amount of expertise with those respective tools. Therefore, it is imperative that VERAView present an interface that is easy to use and easy to understand.

Visualize VERA files.

VERA output (and input) files are **the** source of data for VERAView. VERAView provides a capability to browse, view, and visualize the datasets contained in VERA output file. Future capabilities may add viewing of input files as well.

Analysis.

Essentially, VERAView is an engineering/analysis capability for VERA output. Analysis capabilities will be included to facilitate the user's discovery of critical aspects of the data.

Extensible.

VERAView is designed to be extensible so that visualization widgets and analysis capabilities can be implemented and integrated by users.

3 System Components

Python-2.7 has been chosen as the language and environment for VERAView prototype development. Dependent Python modules are represented below.:

```
VERAView
  |
  +-- wxPython (wxWidgets)
  |
  +-- h5py
  |     |
  +-- numpy
  |
  +-- Pillow
  |
  +-- matplotlib
        |
        +-- pyparsing
        |
        +-- python-dateutil
        |
        +-- six
```

The versions used in the initial distribution are as follows.

Module	Min Version	Used Version
hp5py	2.3.1	2.4.0
matplotlib	1.3.1	1.4.3
numpy	1.8.0	1.9.2
Pillow	2.5.2	2.7.0
pyparsing	2.0.1	2.3.0
python-dateutil	1.5	2.4.1
six	1.4.1	1.9.0
wxPython/wxWidgets	3.0.1.0	3.0.2.0

Through each prototype iteration, VERAView has been tested under Windows, Mac OS X, and Linux.

The wxPython module was chosen for the windowing toolkit. [4] It is a Python wrapper around the wxWidgets cross-platform GUI library. [5] Binary versions of wxPython are available for Windows and Mac OS X, but it must be built from source for most Linux distributions.

The h5py module is a Python wrapper around the HDF5 C application programming interface (API) and library. [6] It is used for all processing of the VERA output file. Since h5py uses numpy, the latter is used for all dataset vector/array manipulations. [7]

Pillow is the Python Imaging Library (PIL) implementation used for image creation in raster-based 2D views. [8] At present, all 2D plots are created using matplotlib. [9]

4 Architecture

Three fundamental elements of the VERAView architecture are the data model, the event system, and the widget framework.

4.1 Data Model

A DataModel class abstracts and encapsulates the HDF5 VERA output file. While exposing the underlying HDF5 groups and datasets, it provides additional functionality in dealing with the VERA data, including:

- determining datasets belonging to each category,
- checking validity,
- mapping axial values to axial level indexes and vice versa,
- determining core symmetry cell ranges,
- calculating and caching values ranges for datasets, and
- mapping between state point indexes and values for state point scalars

As per the VERAOUT specification, DataModel contains an instance of a Core class and a list of State classes instances, the latter representing state point HDF5 groups.

Each state point dataset is examined to determine if it belongs to one or more of the following categories:

- axial
- channel
- detector
- pin
- scalar

A dataset may be "axial" as well as belonging to one of channel, detector, or pin. The dataset shape is used to determine its category(ies).

4.2 Events and User Selection State

A VERA output file may contain many state points, a full core or quarter or eighth symmetry, and multiple axial levels. A 2D view is a slice where all but two index parameters are held constant. Thus, at any point, there is a "state" represent the user's current index selections. This state is shared among all the widgets currently displayed. In addition to GUI controls for selecting certain index values, many widgets also provide a means to interactively select one or more indexes. Each change in the state is propogated to all the widgets and VERAView components via an event. Currently, the state consists of the following properties or attributes:

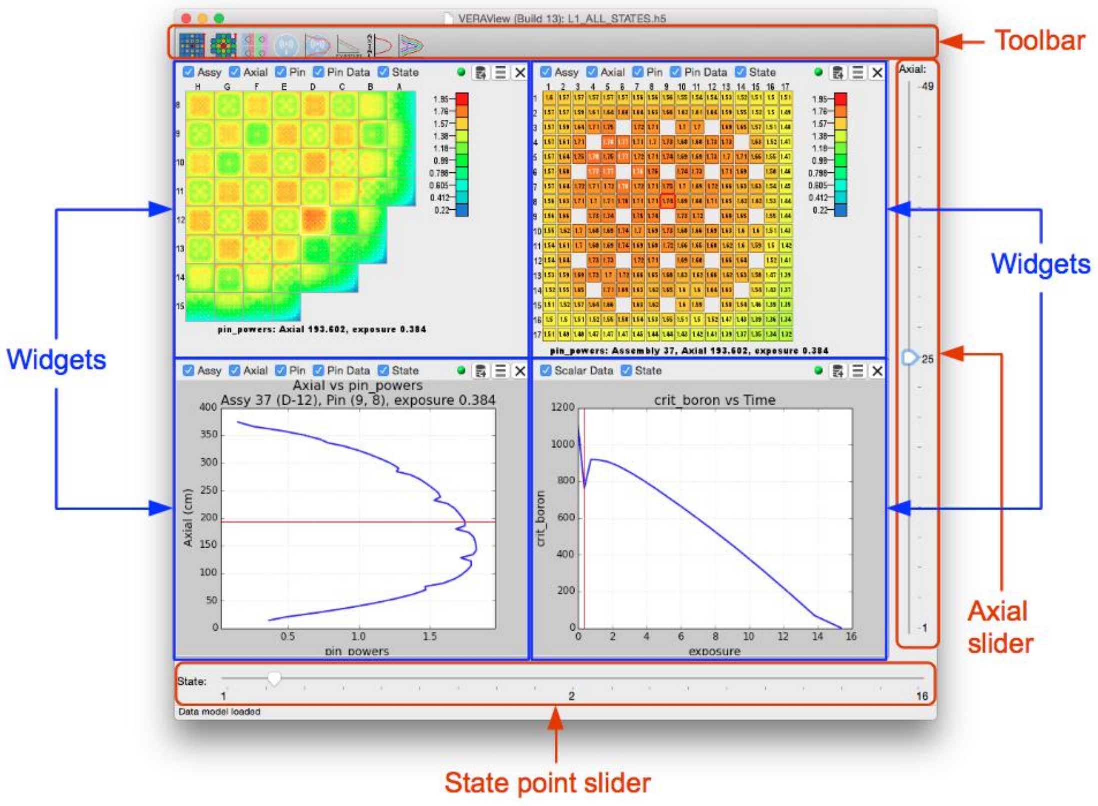
- assembly index
- axial value
- channel dataset
- channel column and row
- detector dataset
- detector index
- pin column and row
- pin dataset
- scalar dataset
- state index
- time dataset

The assembly and detector index attributes are triples consisting of the index (from the core map), the column, and the row. They are tied together such that when the assembly index is changed, the detector index is also changed if there is a detector at the assembly location (column and row). Likewise, when the detector index is changed, the assembly index at the detector column and row is updated.

The axial value is also a triple containing the axial level in cm and the corresponding axial level indexes for pins and detectors, respectively. The channel and pin column and row attributes pairs representing the col, row indexes within the currently selected assembly. The dataset attributes are the names of the currently selected channel, detector, and pin datasets, respectively. The time dataset is the scalar dataset to be used to represent time, currently limited to "exposure", "exposure_efpd", "hours", and the state point index itself. The state index attribute is the state point index.

4.3 Widget Framework

VERAView is implemented as a container or grid of widgets. Refer to the following image.



The image shows a 2x2 grid of four widgets. As the window is resized, the widgets are resized to get equal shares of the window's real estate. Note the axial and state point sliders by which the user can select the axial level and state index, respectively. Individual widgets also provide user actions to select the axial level and state index as described below.

Each widget implementation extends a Widget class (or one of its subclasses) and thus must implement some methods required by the framework. In many cases default behavior can be inherited, but a few methods and properties must be implemented for the widget to function correctly.

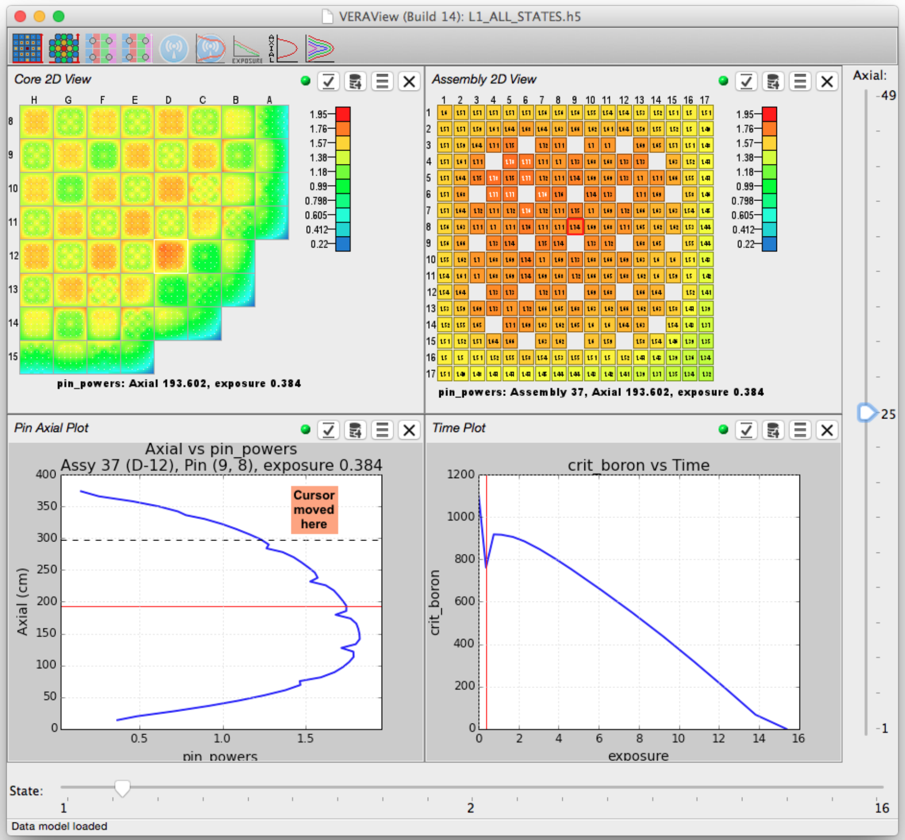
4.3.1 *Event Propogation*

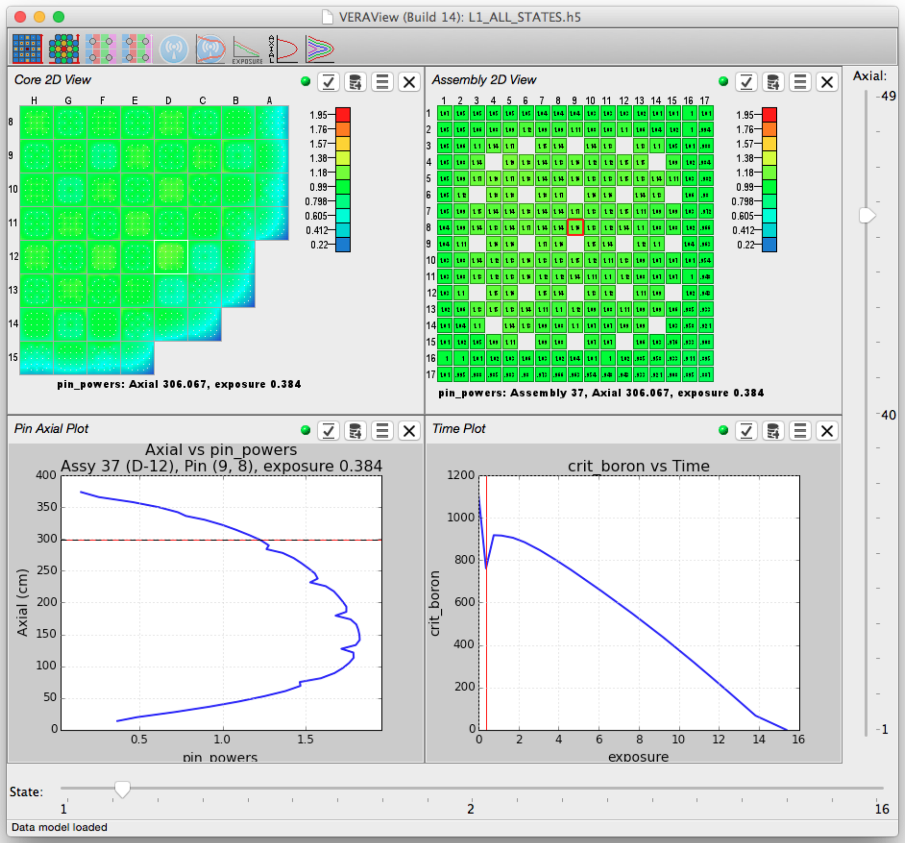
Events are shared among and propogated from widgets based on the checkbox items in the event propogation popup menu accessible via the button at the top of each widget's space. Which event items are available is controlled by the individual widget, and a checked item means the widget shares that state attribute with other widgets that also have the box checked. In the image above, three of the widgets (all but the bottom right) are sharing:

- the assembly index,
- the axial value,
- the pin column and row, and
- the pin dataset.

All widgets are sharing the state point Index, but only the bottom right widget is sharing the scalar dataset.

For example, suppose the user positions the mouse around the axial level of 300 cm in the bottom left widget and clicks. This widget shares the new axial value with the other widgets, as illustrated in the following images.


VERAView (Build 14): L1_ALL_STATES.h5
Core 2D View
Assembly 2D View
pin_powers: Axial 193.602, exposure 0.384
pin_powers: Assembly 37, Axial 193.602, exposure 0.384
Pin Axial Plot
Axial vs pin_powers
Assy 37 (D-12), Pin (9, 8), exposure 0.384
Cursor moved here
Axial (cm)
pin_powers
Time Plot
crit_boron vs Time
crit_boron
exposure
Axial:
State:
Data model loaded


VERAView (Build 14): L1_ALL_STATES.h5
Core 2D View
Assembly 2D View
Axial:
pin_powers: Axial 306.067, exposure 0.384
pin_powers: Assembly 37, Axial 306.067, exposure 0.384
Pin Axial Plot
Time Plot
Axial vs pin_powers
Assy 37 (D-12), Pin (9, 8), exposure 0.384
Axial (cm)
pin_powers
crit_boron vs Time
crit_boron
exposure
State:
Data model loaded

Note that if the "Axial Value" item were not checked in the bottom left widget's event propagation menu, the new axial level selected in the widget would not have been shared or propogated to other widgets. Similarly, were the "Axial Value" item unchecked for the top left widget, it would not have accepted the new axial level.

4.3.2 Widget Toolbar and Menu

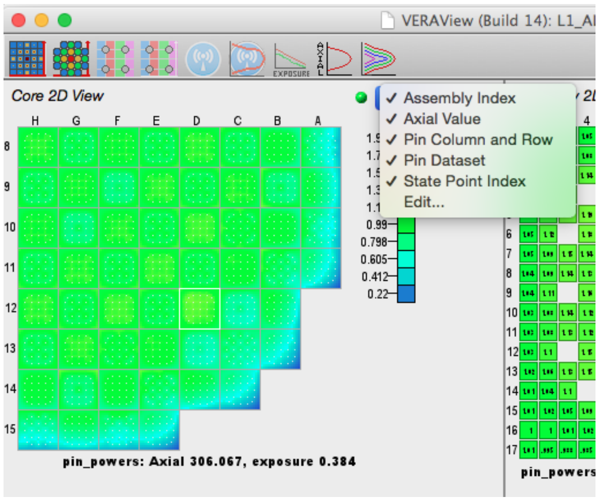
The framework adds a toolbar above each widget containing the widget name, a processing indicator icon, and up to four tool buttons:

- event propogation
- data selection
- widget menu
- close button

Each widget has the option of defining menu items added to a "Save Image" item that's defined for all widgets.

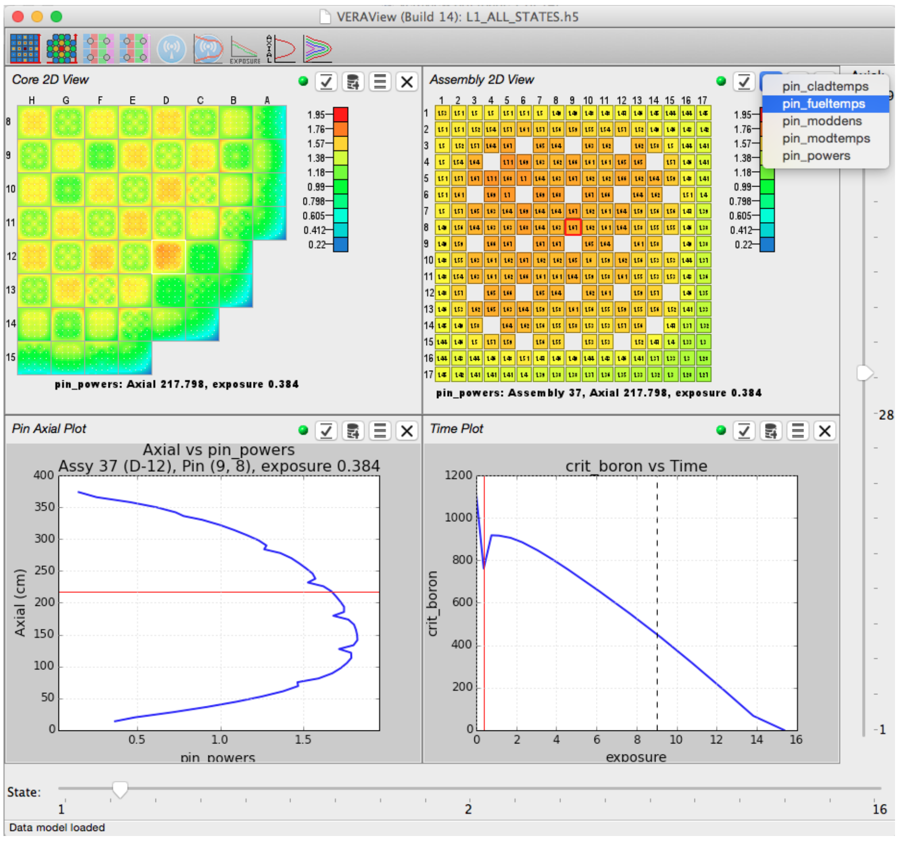
4.3.3 Event Propogation Button

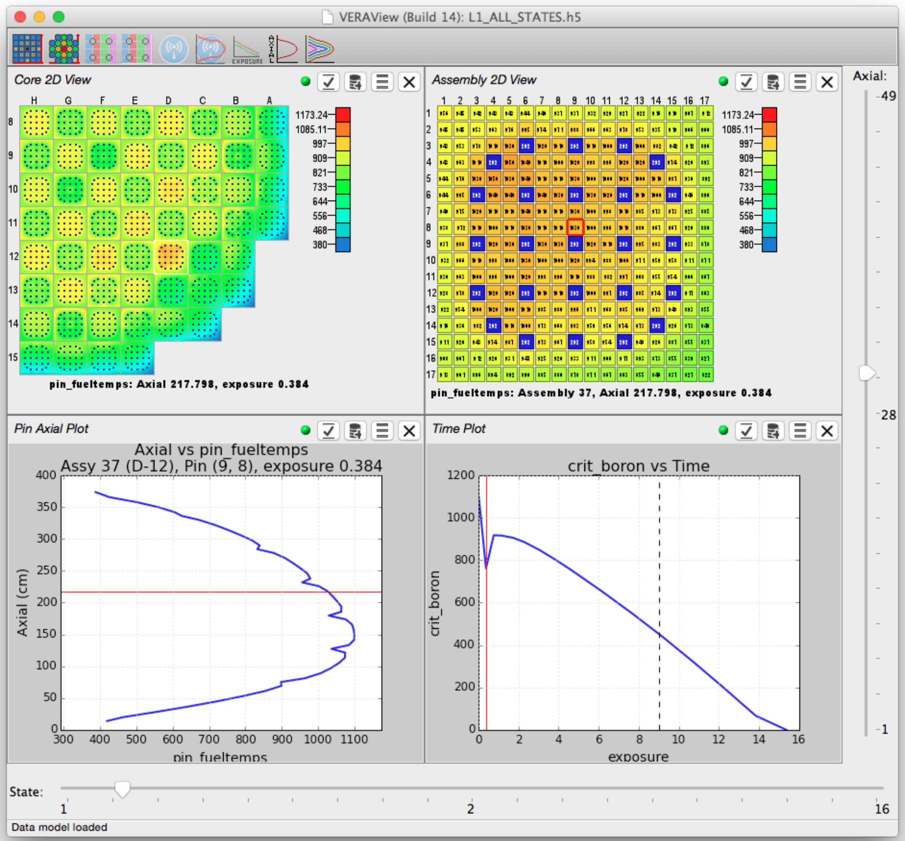
This button has a checkmark icon. When activated, it brings up a popup menu with check items for each of the events for which the widget has registered interest, as described above. There is also an "Edit..." item which brings up a dialog in which multiple events can be (de)selected at one time.



4.3.4 Dataset Selection

Any widget that registers as pertaining to one of the unique dataset categories (channel, detector, pin, scalar) will have an tool button (icon of disk platters with a plus sign). In the images above each includes this button, which brings up a popup menu with all the datasets of that type that were found in the VERA output file. All widgets with checkboxes checked to indicate propogation of the same dataset type will be updated as well as the widget whose button was clicked. Refer to the following images that show dataset selection in the top right widget propogating to all but the bottom right widget.




VERAView (Build 14): L1_ALL_STATES.h5
Core 2D View
Assembly 2D View
pin_fueltemps: Axial 217.798, exposure 0.384
pin_fueltemps: Assembly 37, Axial 217.798, exposure 0.384
Pin Axial Plot
Axial vs pin_fueltemps
Assy 37 (D-12), Pin (9, 8), exposure 0.384
Axial (cm)
pin_fueltemps
Time Plot
crit_boron vs Time
crit_boron
exposure
Axial:
State:
Data model loaded

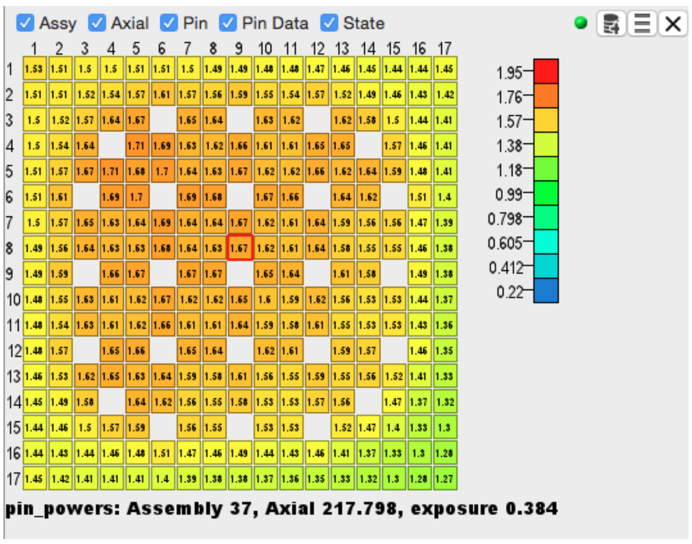
4.3.5 *Adding and Removing Widgets*

Widgets are added by clicking the widget's icon in the VERAView toolbar or choosing from the File->New pullright menu on the menu bar. If necessary, the window size is expanded to add another column in the grid to accomodate the widget. As widgets are closed via the close button in the widget's toolbar, the user may resize the grid by selecting Edit->Resize Grid from the menubar. This brings up a dialog for resizing the grid.

5 Prototype Widgets

Each widget available in the initially distributed VERAView prototype is described below. Note there are a couple of widgets currently under development, including a channel assembly-level 2D view.

5.1 Assembly 2D View



	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
1	1.53	1.51	1.5	1.5	1.51	1.51	1.5	1.49	1.49	1.48	1.48	1.47	1.46	1.45	1.44	1.44	1.45
2	1.51	1.51	1.52	1.54	1.57	1.61	1.57	1.56	1.59	1.55	1.54	1.57	1.52	1.49	1.46	1.43	1.42
3	1.5	1.52	1.57	1.64	1.67		1.65	1.64		1.63	1.62		1.62	1.58	1.5	1.44	1.41
4	1.5	1.54	1.64		1.71	1.69	1.63	1.62	1.66	1.61	1.61	1.65	1.65		1.57	1.46	1.41
5	1.51	1.57	1.67	1.71	1.68	1.7	1.64	1.63	1.67	1.62	1.62	1.66	1.62	1.64	1.59	1.48	1.41
6	1.51	1.61		1.69	1.7		1.69	1.68		1.67	1.66		1.64	1.62		1.51	1.4
7	1.5	1.57	1.65	1.63	1.64	1.69	1.64	1.64	1.67	1.62	1.61	1.64	1.59	1.56	1.56	1.47	1.39
8	1.49	1.56	1.64	1.63	1.63	1.68	1.64	1.63	1.67	1.62	1.61	1.64	1.58	1.55	1.55	1.46	1.38
9	1.49	1.59		1.66	1.67		1.67	1.67		1.65	1.64		1.61	1.58		1.49	1.38
10	1.48	1.55	1.63	1.61	1.62	1.67	1.62	1.62	1.65	1.6	1.59	1.62	1.56	1.53	1.53	1.44	1.37
11	1.48	1.54	1.63	1.61	1.62	1.66	1.61	1.61	1.64	1.59	1.58	1.61	1.55	1.53	1.53	1.43	1.36
12	1.48	1.57		1.65	1.66		1.65	1.64		1.62	1.61		1.59	1.57		1.46	1.35
13	1.46	1.53	1.62	1.65	1.63	1.64	1.59	1.58	1.61	1.56	1.55	1.59	1.55	1.56	1.52	1.41	1.33
14	1.45	1.49	1.58		1.64	1.62	1.56	1.55	1.58	1.53	1.53	1.57	1.56		1.47	1.37	1.32
15	1.44	1.46	1.5	1.57	1.59		1.56	1.55		1.53	1.53		1.52	1.47	1.4	1.33	1.3
16	1.44	1.43	1.44	1.46	1.48	1.51	1.47	1.46	1.49	1.44	1.43	1.46	1.41	1.37	1.33	1.3	1.28
17	1.45	1.42	1.41	1.41	1.41	1.4	1.39	1.38	1.38	1.37	1.36	1.35	1.33	1.32	1.3	1.28	1.27

pin_powers: Assembly 37, Axial 217.798, exposure 0.384

This widget provides a 2D raster view of the pins in the currently selected assembly. It displays pin datasets and represents the pin column and row state attribute by highlighting that pin with a red border. Values at each pin for the current pin dataset are represented by background color, and the values in limited precision are displayed. It is a zoomable widget, meaning a left-click drag rubber band box may be drawn around a portion of the widget to cause a zoom to the selected array of pins. Selecting "unzoom" from the widget menu button or the context menu (right-click in the widget) restores to the previous zoom level. Left-clicking a pin selects a new pin column and row state value, which is propogated if the "Pin Column and Row" event menu item is checked.

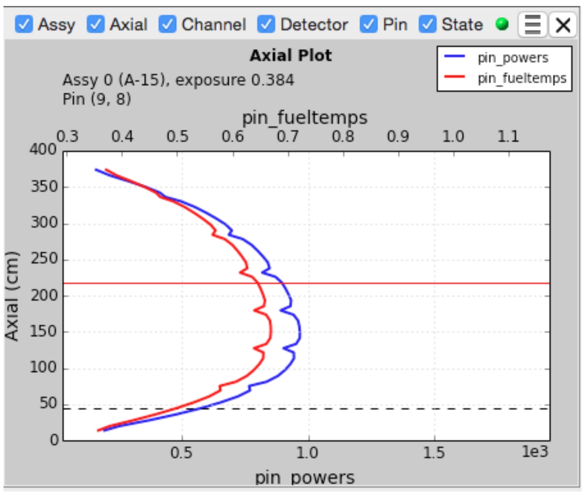
Event attributes received:

- assembly index
- axial value
- pin column and row
- pin dataset
- state index
- time dataset

Event attributes sent:

- pin column and row
- pin dataset

5.2 Axial Plot



This widget is a 2D plot of one or more axial datasets. The datasets to plot are chosen by selecting "Select Datasets" from the widget menu button and interacting with the resulting DataSet Chooser dialog. Each dataset that has been identified as being of type axial is available for display, and multiple datasets may be plotted at once. Detector dataset are plotted as points, but all other datasets are plotted as lines. However, there are only two axes for scaling, the bottom and top. Thus, a scale value may be entered to scale a dataset that is not the basis for one of the axes. The current axial value is represented with a red horizontal line. A dotted black horizontal line follows mouse movement in the widget.

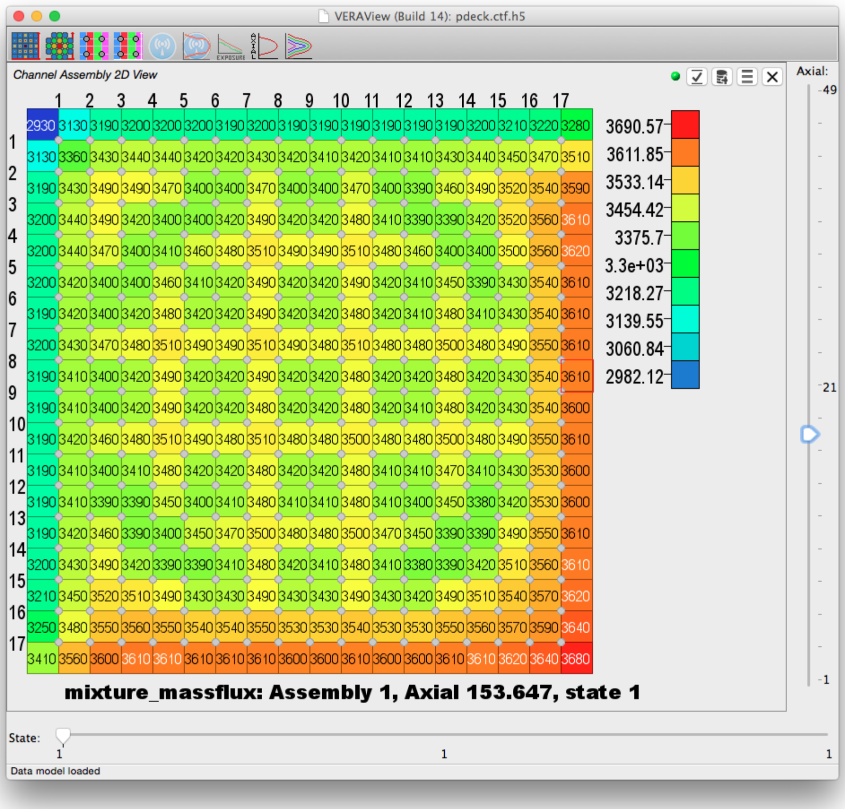
Event attributes received:

- assembly index
- axial value
- channel dataset
- detector dataset
- pin dataset
- state index
- time dataset

Event attributes sent:

- axial value

5.3 Channel Assembly 2D View



2930	3130	3190	3200	3200	3200	3190	3200	3190	3190	3190	3190	3190	3190	3200	3210	3220	3280
3130	3360	3430	3440	3440	3420	3420	3430	3420	3410	3420	3410	3410	3430	3440	3450	3470	3510
3190	3430	3490	3490	3470	3400	3400	3470	3400	3400	3470	3400	3390	3460	3490	3520	3540	3590
3200	3440	3490	3420	3400	3400	3420	3490	3420	3420	3480	3410	3390	3390	3420	3520	3560	3610
3200	3440	3470	3400	3410	3460	3480	3510	3490	3490	3510	3480	3460	3400	3400	3500	3560	3620
3200	3420	3400	3400	3460	3410	3420	3490	3420	3420	3490	3420	3410	3450	3390	3430	3540	3610
3190	3420	3400	3420	3480	3420	3420	3490	3420	3420	3480	3420	3410	3480	3410	3430	3540	3610
3200	3430	3470	3480	3510	3490	3490	3510	3490	3480	3510	3480	3480	3500	3480	3490	3550	3610
3190	3410	3400	3420	3490	3420	3420	3490	3420	3420	3480	3420	3420	3480	3420	3430	3540	3610
3190	3410	3400	3420	3490	3420	3420	3480	3420	3420	3480	3420	3410	3480	3420	3430	3540	3600
3190	3420	3460	3480	3510	3490	3480	3510	3480	3480	3500	3480	3480	3500	3480	3490	3550	3610
3190	3410	3400	3410	3480	3420	3420	3480	3420	3420	3480	3410	3410	3470	3410	3430	3530	3600
3190	3410	3390	3390	3450	3400	3410	3480	3410	3410	3480	3410	3400	3450	3380	3420	3530	3600
3190	3420	3460	3390	3400	3450	3470	3500	3480	3480	3500	3470	3450	3390	3390	3490	3550	3610
3200	3430	3490	3420	3390	3390	3410	3480	3420	3410	3480	3410	3380	3390	3420	3510	3560	3610
3210	3450	3520	3510	3490	3430	3430	3490	3430	3430	3490	3430	3420	3490	3510	3540	3570	3620
3250	3480	3550	3560	3550	3540	3540	3550	3530	3530	3540	3530	3530	3550	3560	3570	3590	3640
3410	3560	3600	3610	3610	3610	3610	3610	3600	3600	3610	3600	3600	3610	3610	3620	3640	3680

mixture_massflux: Assembly 1, Axial 153.647, state 1

This widget provides a 2D raster view of the channels in the currently selected assembly. It displays channel datasets and represents channel pin column and row state attribute by highlighting that channel with a red border. Values in each channel cell for the current channel dataset are represented by background color, and the values in limited precision are displayed. It is a zoomable widget, meaning a left-click drag rubber band box may be drawn around a portion of the widget to cause a zoom to the selected array of channels. Selecting "unzoom" from the widget menu button or the context menu (right-click in the widget) restores to the previous zoom level. Left-clicking a channel cell selects a new channel column and row state value, which is propogated if the "Channel Column and Row" event menu item is checked.

Pins are represented in the image to orient the channel locations, but pin display can be toggled via the widget menu.

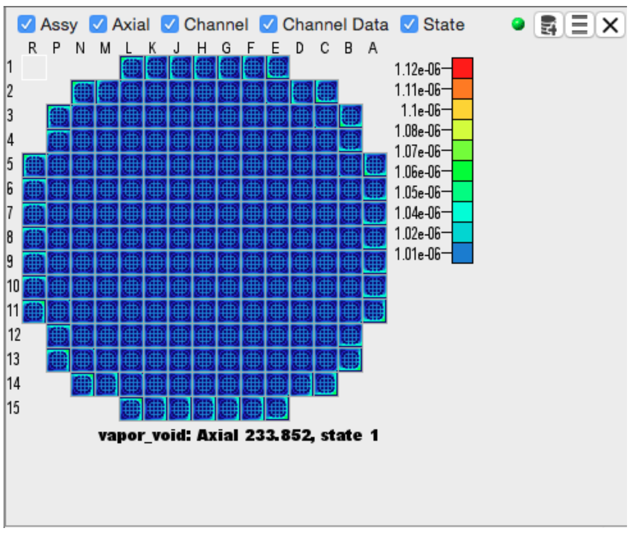
Event attributes received:

- assembly index
- axial value
- channel column and row
- channel dataset
- state index
- time dataset

Event attributes sent:

- channel column and row
- channel dataset

5.4 Channel 2D View



This widget provides a 2D raster view of channels in the core. It highlights the currently selected assembly (assembly index state attribute) with a white border. It displays channel datasets, and values for each channel in each assembly for the current channel dataset are represented by color. It is also a zoomable widget. When zoomed down to a single assembly, channel values are displayed with circles indicating pin locations. Zoom operations are the same as for the Assembly 2D View widget. Left-clicking an assembly selects it, and the new assembly index is propogated if the "Assembly Index" event menu item is checked. Also, the channel location clicked within the assembly is propogated as the channel column and row state attribute if the "Channel Column and Row" event menu item is checked.

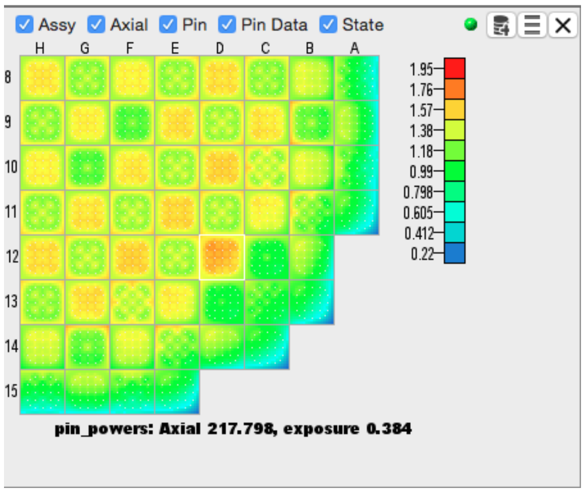
Event attributes received:

- assembly index
- axial value
- channel column and row
- channel dataset
- state index
- time dataset

Event attributes sent:

- assembly index
- channel column and row
- channel dataset

5.5 Core 2D View



pin_powers: Axial 217.798, exposure 0.384

This widget provides a 2D raster view of the assemblies in the core. It highlights the currently selected assembly (assembly index state attribute) with a white border. It displays pin datasets, and values at each pin in each assembly for the current pin dataset are represented by color. It is also a zoomable widget. When zoomed down to a single assembly, pins in that assembly are represented as circles. Zoom operations are the same as for the Assembly 2D View widget. Left-clicking an assembly selects it, and the new assembly index is propogated if the "Assembly Index" event menu item is checked. Also, the pin clicked within the assembly is propogated as the pin column and row state attribute if the "Pin Column and Row" menu item is checked.

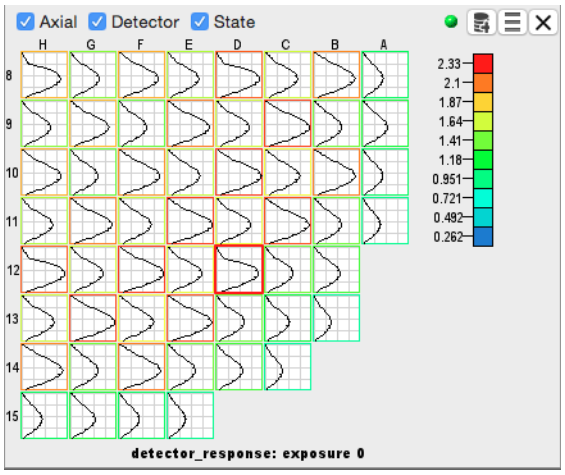
Event attributes received:

- assembly index
- axial value
- pin column and row
- pin dataset
- state index
- time dataset

Event attributes sent:

- assembly index
- pin column and row
- pin dataset

5.6 Detector 2D View



This widget provides a 2D view of the detectors in the core at assembly locations where detectors exist. It highlights the currently selected detector (detector index state attribute) with a red border. It displays detector datasets as an axial plot within each detector location. It is also a zoomable widget. Zoom operations are the same as for the Assembly 2D View widget. Left-clicking an assembly selects the detector at that location, and the new detector index is propogated if the "Detector Index" event menu item is checked.

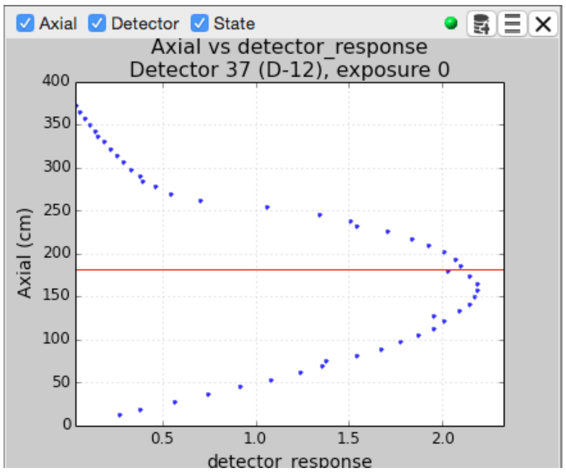
Event attributes received:

- axial value
- detector dataset
- detector index
- state index
- time dataset

Event attributes sent:

- detector dataset
- detector index

5.7 Detector Axial Plot



This widget plots points for detector dataset values against axial value for the current detector index and state index attributes. It will be subsumed by the Axial Plot widget once the latter is fully functional in using the detector dataset attribute. For now, the utility of this widget is response to changes in the detector dataset attribute. The current axial value is represented with a red horizontal line, and a dotted black horizontal line follows mouse movement in the widget.

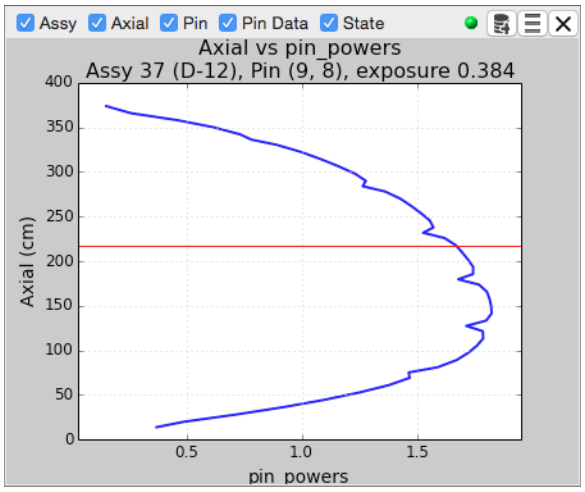
Event attributes received:

- axial value
- detector dataset
- detector index
- state index

Event attributes sent:

- axial value
- detector dataset

5.8 Pin Axial Plot



This widget plots lines for pin dataset values against axial value for the current assembly index, pin column and row, and state index state attributes. It will be subsumed by the Axial Plot widget once the latter is fully functional in using the pin dataset attribute. For now, the utility of this widget is response to changes in the pin dataset attribute. The current axial value is represented with a red horizontal line, and a dotted black horizontal line follows mouse movement in the widget.

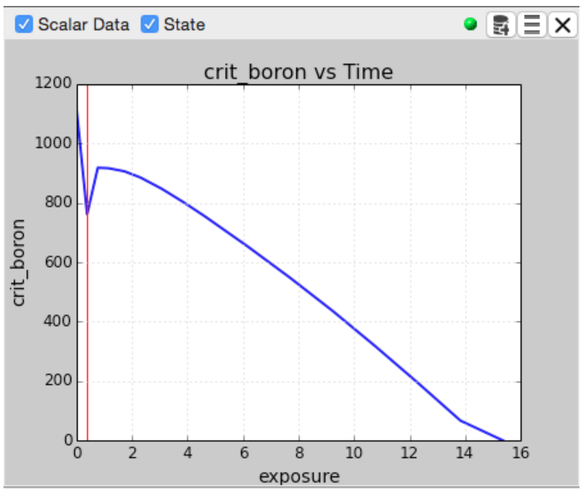
Event attributes received:

- assembly index
- axial value
- pin column and row
- pin dataset
- state index

Event attributes sent:

- axial value
- pin dataset

5.9 Time Plot



This widget plots scalar datasets against time. The current state point (time) value is represented with a red vertical line, and a dotted black vertical line follows mouse movement. Clicking within the plot sets the state index attribute.

Event attributes received:

- scalar dataset
- state index

Event attributes sent:

- state index

6 Notes

1 Godfrey, A., G. Davidson, B. Collins, S. Palmtag, VERAOUT - VERA HDF5 Output Specification, 2 May 2014, CASL-U-2014-0043-001.

2 http://www.paraview.org/.

3 https://visit.llnl.gov/.

4 http://wxpython.org/.

5 https://www.wxwidgets.org/.

6 http://www.h5py.org/.

7 http://www.numpy.org/.

8 https://python-pillow.github.io/.

9 http://matplotlib.org/.